Metric and Appearance Based Visual SLAM for Mobile Robots

by

Caner Şahin

Submitted to the Graduate School of Sabancı University
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

August 2013

Metric and Appearance Based Visual SLAM for Mobile Robots

APPROVED BY:

Prof. Dr. Mustafa Ünel
(Thesis Advisor) .....

Assoc. Prof. Ali Koşar .....

Assoc. Prof. Gözde Ünal .....

Assist. Prof. Hakan Erdoğan .....

Assist. Prof. Hüseyin Üvet .....

DATE OF APPROVAL: 05/08/2013

Metric and Appearance Based Visual SLAM for Mobile Robots

Caner Şahin

ME, Master's Thesis, 2013

Thesis Supervisor: Prof. Dr. Mustafa Ünel

Keywords: Visual SLAM, Navigation, Wheeled Mobile Robot, Visual Sensor

Abstract

Simultaneous Localization and Mapping (SLAM) maintains autonomy for mobile robots and it has been studied extensively during the last two decades. It is the process of building the map of an unknown environment and determining the location of the robot using this map concurrently. Different kinds of sensors such as Global Positioning System (GPS), Inertial Measurement Unit (IMU), laser range finder and sonar are used for data acquisition in SLAM. In recent years, passive visual sensors are utilized in visual SLAM (vSLAM) problem because of their increasing ubiquity.

This thesis is concerned with the metric and appearance-based vSLAM problems for mobile robots. From the point of view of metric-based vSLAM, a performance improvement technique is developed. Template matching based video stabilization and Harris corner detector are integrated. Extracting Harris corner features from stabilized video consistently increases the accuracy of the localization. Data coming from a video camera and odometry are fused in an Extended Kalman Filter (EKF) to determine the pose of the robot and build the map of the environment. Simulation results validate the performance improvement obtained by the proposed technique. Moreover, a visual perception system is proposed for appearance-based vSLAM and used for under vehicle classification. The proposed system consists of three main parts: monitoring, detection and classification. In the first part a new catadioptric camera system, where a perspective camera points downwards to a convex mirror mounted to the body of a mobile robot, is designed. Thanks to the catadioptric mirror the scenes against the camera optical axis direction can be viewed. In the second part speeded up robust features (SURF) are used to detect the hidden objects that are under vehicles. Fast appearance

based mapping algorithm (FAB-MAP) is then exploited for the classification of the means of transportations in the third part. Experimental results show the feasibility of the proposed system. The proposed solution is implemented using a non-holonomic mobile robot. In the implementations the bottom of the tables in the laboratory are considered as the under vehicles. A database that includes different under vehicle images is used. All the algorithms are implemented in Microsoft Visual C++ and OpenCV 2.4.4.

Mobil Robotlar İçin Metrik ve Görünüm Tabanlı Görsel Eş Zamanlı Konumlama ve Haritalama

Caner Şahin

ME, Master Tezi, 2013

Tez Danışmanı: Prof. Dr. Mustafa Ünel

Anahtar Kelimeler: Görsel Eş Zamanlı Konumlama ve Haritalama, Navigasyon, Tekerlekli Mobil Robot, Görsel Sensör

Özet

Eş Zamanlı Konumlama ve Haritalama (EZKH) mobil robotlarda otonomiyi sağlamakta ve son yirmi yıldır kapsamlı olarak çalışılmaktadır. EZKH bilinmeyen bir ortamın haritasının çıkartılması ve bu haritanın robot pozisyonunu hesaplamak için eş zamanlı olarak kullanılmasıdır. Global Pozisyonlama Sistemi (GPS), Atalet Ölçüm Ünitesi, lazer mesafe ölçme cihazı veya sonar gibi çeşitli sensörler veri toplamak için EKZH' de kullanılmaktadır. Son zamanlarda, pasif görsel sensörler kullanımındaki artıştan dolayı görsel EKZH (gEKZH) probleminde faydalanılmaktadır.

Bu tez, mobil robotlar için metrik ve görünüş tabanlı gEKZH problemleri ile ilgilenmektedir. Metrik tabanlı gEKZH açısından bir performans iyileştirme tekniği geliştirilmiştir. Şablon eşleme tabanlı video stabilizasyonu ve Harris köşe sezicisi bütünleştirilmektedir. Tutarlı olarak stabilize edilmiş videodan Harris köşe özniteliklerinin çıkarımı konumlama doğruluğunu artırmaktadır. Video kamera ve odometreden gelen datalar mobil robotun duruşunu hesaplamak ve ortamın haritasını çıkarmak için genişletilmiş Kalman filtresinde tümleştirilmektedir. Simulasyon sonuçları önerilen teknikle elde edilen performans iyileştirmesini doğrulamaktadır. Ayrca, görünüş tabanlı gEKZH algoritması için bir görsel algılama sistemi önerilmektedir ve araçların sınıflandırılması için kullanılmaktadır. Önerilen sistem üç ana kısımdan oluşur: görüntüleme, sezme ve sınıflandırma. Birinci kısımda perspektif kameranın mobil robotun gövdesine bağlanan konveks aynaya doğru hizalanmış olduğu bir katadioptrik kamera sistemi geliştirilmiştir. Katadioptrik ayna sayesinde kamera optik ekseni yönünün tersindeki alanlar görüntülenebilmek-

tedir. İkinci kısımda araçların altındaki gizlenmiş nesneleri sezmek için Hızlandırılmış Gürbüz Öznitelikler (HGÖ) kullanılmaktadır. Hızlı görünüş tabanlı haritalama algoritmasından araçları sınıflandırmak için üçüncü bölümde yararlanılmıştır. Deneysel sonuçlar önerilen sistemin uygulanabilirliğini göstermektedir. Önerilen çözüm bir holonomik olmayan mobil robot kullanılarak uygulanmıştır. Uygulamalarda, laboratuar ortamında bulunan masaların alt kısımları araç alt gövdeleri olarak düşünülmüştür ve farklı araç altı görüntülerinden oluşan bir veritabanı kullanılmıştır. Tüm algoritmalar Microsoft Visual C++ ve OpenCV 2.4.4 kütüphanelerinde gerçeklenmiştir.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Dr. Mustafa Ünel. He has everything a good supervisor should have. His wise words at the right times, hours we spent together in his office 1089, the freedom to explore interesting things. He has shown great faith in my work when at times I have felt none for it myself.

I would like to thank Assoc. Prof. Ali Koşar, Assist. Prof. Hakan Erdoğan, Assoc. Prof. Gözde Ünal and Assist. Prof. Hüseyin Üvet for their feedbacks and spending their valuable time to serve as my jurors.

I am deeply grateful to my beloved parents, Ayşe & Sabri Şahin for raising me with great love. Their priceless trust in me and their support in every instant of my life cannot be compared to anything else.

Finally, I would like to thank my brothers Cüneyt & Cemil Şahin for all their motivation and support throughout my life.

Contents

List of Figures

List of Tables

Chapter I

1 Introduction

If you are a pilot of an aeroplane flying in the sky you may want to know where you are. You might be a traveller in your car having a long journey and be curious about the remaining kilometers to the destination. In summer, having a holiday in the middle of the Mediterranean with your yacht may require the latitude and longitude data according to the Greenwich. Also, a mobile robot can be launched to Mars by a general of an army and a radar can tell the position of the mobile robot. Moreover, if you are a researcher in robotics community you may want to navigate your mobile robot without noticing if it is an unmanned aerial vehicle, autonomous underwater vehicle or non-holonomic wheeled mobile robot. In all kinds of applications that are mentioned above one must answer the question "How can I solve navigation problem?". The main aim of all navigation processes including dead reckoning, pilotage, celestial navigation, radio navigation, radar navigation and satellite navigation etc. is the determination of a navigator's exact position and direction with respect to a fixed reference frame.

In mobile robotics applications simultaneous localization and mapping (SLAM) is one of the methods that is used for navigation. It provides autonomy for mobile robots and has been studied extensively during the last two decades [3–9]. SLAM is concerned with the ability of an autonomous vehicle

to navigate through an unknown environment and incrementally build a map of the environment while localising itself within this map. At the beginning the map of the environment and the mobile robot position are not known and the mobile robot starts to navigate in the environment which has features that are artificial or natural. The SLAM process comprises finding appropriate representation for both the observation and motion model [1]. In order to do so, different kinds of sensors mounted to the mobile robot are utilized. Sonar, laser, inertial and vision based sensors are most commonly used sensors in SLAM applications for data acquisition. Sonar based systems are fast and relatively cheap but they tend to produce less accurate and more noisy measurements compared to other kinds of sensors [10]. Laser range finders achieves significant improvements over the ultrasonic range sensors (sonar) owing to the use of laser light instead of sound, but their point-to-point measurement characteristic is limited [10]. Inertial sensors such as odometers and inertial measurement units (IMU) are extremely sensitive to measurement errors. On the other hand, vision-based systems are passive, they have long range and high resolution. One can extract features nearly at infinity using a visual sensor. In recent years, because of the increasing ubiquity of passive vision - based systems, there has been intense research into visual SLAM (vSLAM) using primarily standard perspective cameras. Another reason why cameras are being used in SLAM process is that the acquired data from images can be directly applicable in the existing computer vision techniques for extracting and matching visual features [11]. Despite the fact that the computational load of the applied algorithms in computer vision exceeds capacities of typical embedded computers and must be considered in real-time applications, many of the most successful visual SLAM implementations uti-

lize state-of-the-art feature extraction and matching, structure from motion and scene modelling computer vision techniques. Moreover outputs of the vision algorithms facilitate the solution of the challenging problems such as loop closure, data association etc. This fact is one of the main motivations for using vision sensors.

1.1 Objective

The main goal of this thesis is to improve the performance of a metric based vSLAM algorithm and utilize the fast appearance based mapping (FAB-MAP) algorithm for detecting and classifying objects that are mounted to the under frames of the means of transportations. A non-holonomic mobile robot and its kinematic model are used in implementations including both simulations and experimental work.

This thesis concerns with the performance improvement in vSLAM where the map of the environment is built with metric data. When a non-holonomic wheeled mobile robot (WMR) navigates in an unknown environment, some undesired phenomena such as vibrations on the mobile robot and the speed bump constructions in the environment might occur. To prevent the negative effects caused by such events, a novel method is presented. It enhances the consistency of the constructed map using stabilized corner features. The proposed method integrates template matching based video stabilization and Harris corner detector. Also, extracting Harris corner features from stabilized video consistently increases the accuracy of the localization. Data coming from a video camera and odometry are fused in an Extended Kalman Filter (EKF) to determine the pose of the robot and build the map of the environment. Moreover, imaging and classification of under vehicles are provided

and hidden objects are detected. In order to do so, a solution consisting of three main parts is proposed: monitoring, detection and classification. In the first part a new catadioptric camera system in which the perspective camera points downwards to the catadioptric mirror mounted to the body of a mobile robot is designed. Thanks to the catadioptric mirror the scenes against the camera optical axis direction can be viewed. In the second part speeded up robust features (SURF) are used in an object recognition algorithm. Fast appearance based mapping algorithm (FAB-MAP) is exploited for the classification of the means of transportations in the third part.

1.2 Structure of the Thesis and Contributions

The rest of this thesis is organized as follows:

Chapter 2 reviews the most commonly used SLAM techniques for the navigation of a mobile robot. Particular attention is devoted to Kalman filter and appearance based SLAM algorithms since they will be used in subsequent chapters. Furthermore, most common computer vision methods and a variety of visual features used in vSLAM are described.

Chapter 3 presents an improvement technique for the performance of the vSLAM algorithm. In this technique, odometry and standard perspective camera data are fused in an extended Kalman filter (EKF) to determine the pose of the robot and build the map of the environment. Template matching based video stabilization and Harris corner detector are integrated to increase the consistency of the map building and the localization.

Chapter 4 proposes an imaging and classification framework for under vehicles using object detection and the fast appearance based mapping (FAB-MAP) algorithm where a catadioptric camera is utilized.

Chapter 5 describes the simulation results obtained from the work presented in Chapter 3 and experimental results for Chapter 4.

Chapter 6 concludes the thesis and indicates possible future directions.

Contributions of the thesis can be summarized as follows:

- A performance improvement technique for vSLAM that extracts stabilized Harris corner features using template matching based stabilized video sequences is proposed. Stabilized feature extraction ensures both consistency in map building and localization of the mobile robot.
- A new under vehicle perception system is developed for high level safety measures using appearance based vSLAM.
- The perception system classifies means of transportations via FAB-MAP algorithm and an object recognition algorithm is used to detect hidden objects.

Chapter II

2 Background

An autonomous mobile robot equipped with sensors is being used to achieve a variety of tasks in different environments that have randomly distributed landmarks. Throughout the processes control commands are being sent to the actuators utilizing the kinematic model of the mobile robot. Different kinds of sensors acquire data in the environment and they should be fused in sensor fusion algorithms. Approximations for the motion model of the mobile robots, inaccurate control commands, noisy and biased sensor readings make the realization of the tasks unreliable. Moreover, we want the mobile robot achieve the tasks fully autonomously and navigate in the environment in a reliable manner. Leonard and Durrant-Whyte [4] summarized the general problem of mobile robot navigation by three questions:

- Where am I?
- Where am I going?
- How should I get there?

SLAM has been one of the main research topics to ensure autonomy and reliable navigation in the mobile robotics. When a mobile robot navigates in an environment, it is hard to compute the deterministic value of the robot

pose and landmark positions. We estimate their approximate values. Hence, in this chapter we will firstly review the SLAM problem in a probabilistic framework and then explain the required material for the rest of the thesis.

2.1 SLAM Techniques

2.1.1 Formulation of the SLAM Problem

SLAM is the process of building the map of an unknown environment and determining the location of the robot using this map concurrently. In SLAM, the position of the mobile robot and the location of the landmarks that are used to represent the map are estimated without any a priori knowledge of location [1]. Assume that a non-holonomic wheeled mobile robot is navigating through an unknown environment and taking measurements from a number of unknown landmarks with a sensor (laser range finder or sonar) as shown in Figure 2.1. In robotics, a non-holonomic system has less controllable degrees of freedom than the total degrees of freedom. At a given time instant k, the following quantities are defined:

- $\mathbf{x}_k$: a state vector describes the pose of the mobile robot
- $\mathbf{u}_k$: a control vector, applied to the mobile robot at time *k-1* to drive the vehicle to the state $\mathbf{x}_k$ at time k
- $\mathbf{m}_i$: a vector involves the position of the ith feature whose true location is fixed.
- $\mathbf{z}_{ik}$: an observation for the location of the ith feature at time k.

If $\mathbf{X}_{0:k}$ is defined as the history of vehicle locations, $\mathbf{U}_{0:k}$ is the history of control inputs, $\mathbf{m}$ is the set of all landmarks and $\mathbf{Z}_{0:k}$ is the set of all feature

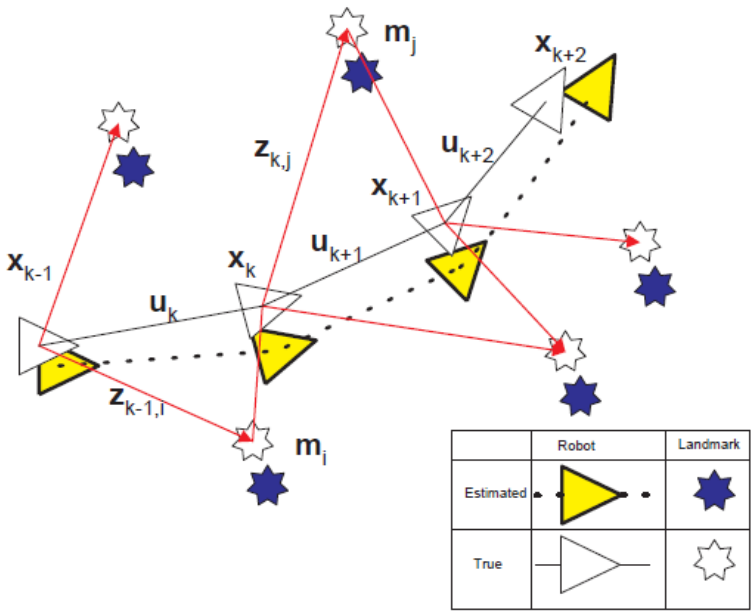


Figure 2.1: The SLAM problem [1]

observations, then SLAM problem can be formulated as a recursive Bayesian estimation problem [1]:

$$P(\mathbf{x_k}, \mathbf{m} \mid \mathbf{Z_{0:k}}, \mathbf{U_{0:k}}, \mathbf{x_0}) = \frac{P(\mathbf{z_k} \mid \mathbf{x_k}, \mathbf{m}) P(\mathbf{x_k}, \mathbf{m} \mid \mathbf{Z_{0:k-1}}, \mathbf{U_{0:k}}, \mathbf{x_0})}{P(\mathbf{z_k} \mid \mathbf{Z_{0:k-1}}, \mathbf{U_{0:k}})} \quad (2.1)$$

Note that depending on the assumptions made about the probability density functions, this formulation may imply the map building or localization problem and may lead to the different SLAM algorithms. For example, the map building problem may be formulated as computing the conditional density:

$$P(\mathbf{m} \mid \mathbf{X_{0:k}}, \mathbf{Z_{0:k}}, \mathbf{U_{0:k}}) \quad (2.2)$$

This density function assumes that the pose of the mobile robot is known up to and including time instant k. The map of the environment can be

calculated using the robot location data. Similarly, the localization problem may be formulated as computing the conditional density:

$$P(\mathbf{x_k} \mid \mathbf{Z_{0:k}}, \mathbf{U_{0:k}}, \mathbf{m}) \tag{2.3}$$

This density function assumes that the map of the environment is known and the pose of the mobile robot is estimated with respect to the built map.

SLAM problem can be adapted to the systems that use visual sensors to take observations. Using the visual sensors, natural or artificial landmarks are extracted, matched and tracked between consecutive video frames. A captured image can be described as $I(x, y, t)$ where (x, y) is the location of a pixel which has an intensity value and t is the acquisition time [12]. Suppose that there are displacements $d=(\xi,\eta)$ and the time difference between two consecutive frames τ is small. The relation between these two images is expressed as the following equation [2]:

$$I(x, y, t + \tau) = I(x - \xi, y - \eta, t) \tag{2.4}$$

In Eq. (2.4) the displacements are related to the movement of the visual sensor. If the displacement of a visual feature is estimated, then the movement of the visual sensor can be calculated. These visual features can be extracted using Harris corner detector, Fast Corner detector, SIFT or SURF. Extracted features can be further used for the tracking of feature point positions continuously [2] and permits the concept of vSLAM (Figure 2.2).

The visual sensors also provide range, bearing and elevation information and can be fused with other kinds of sensors such as laser, sonar or inertial sensors. In order to fuse visual sensors, output of other sensors are combined

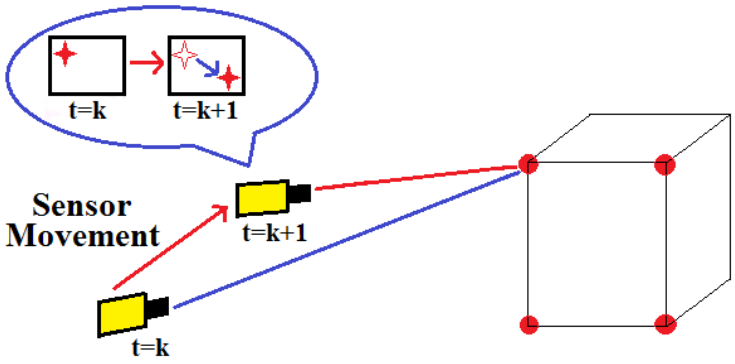


Figure 2.2: The concept of vSLAM [2]

with visual sensor data in SLAM process as shown in Figure 2.3. Navigation information (position, velocity, attitude etc.) and errors in sensors are estimated by integrating information from visual and other sensors. Assuming feature points are time invariant in the local coordinate frame, navigation errors come from mainly sensor outputs. Thus, by compensating estimated errors from sensor output, navigation data can be precisely calculated [2].

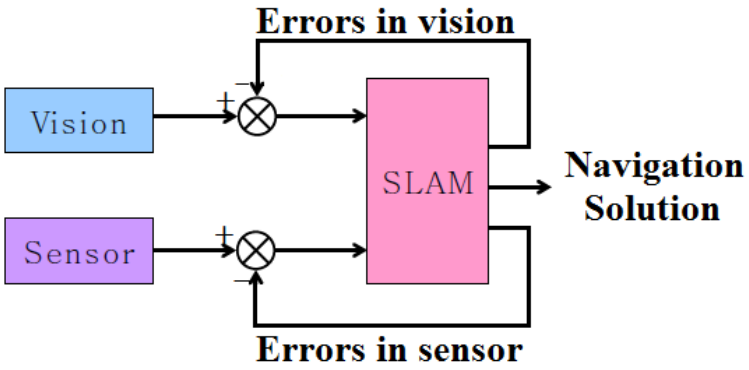


Figure 2.3: Sensor fusion in vSLAM [2]

2.1.2 Kalman Filter Based SLAM

Proposed solutions to the SLAM problem using Kalman filter (KF) approximates the joint posterior in Eq. (2.1) in the form of a state space model with additive Gaussian noise. For both observation and motion model appropriate representations must be found. In [1] EKF based SLAM algorithm is summarized.

The inception of the EKF-SLAM algorithm occured in 1986 and is due to Smith and Cheeseman [13]. They proposed an estimation method for the uncertainty of a frame relative to another and show how the reduction in uncertainty due to sensing can be mapped into any frame. A mobile robot is used to show the estimation of uncertainties in three degrees of freedom *(x, y, θ)*. A simple EKF is used in sensor fusion algorithm.

In the prominent paper of Smith, Self and Cheeseman [14] the landmark based mapping is used for the representation of the environment and EKF formulation of SLAM is described. The vehicle and landmark covariances are approximated by Gaussian distributions. To improve the accuracy of the map building process, a GUI is developed for a mobile robot in [15]. The proposed method enables the operator of the mobile robot to compare the built map using different sensors with the video camera frames. Laser range finder is used for observation and EKF is used to improve the self-localization of the mobile robot. In a separate work, CCD camera and sonar sensors are fused in EKF [8] to enhance the reliability and precision of the environment observations used for the SLAM. Hough transform is applied to the data both acquired from sonar and vision sensors. Due to a multi-sensor system, composed of laser range finder and monocular camera, weighted least square fitting and Canny operator are used to extract two dimensional features and

vertical edges to be utilized in EKF-SLAM [16]. In the paper of Karlsson et al. [16] they aim to handle dynamic changes in the environment such as lighting changes, moving objects and/or people. In order to do so, vision and odometry based vSLAM algorithm allows low cost navigation in cluttered and populated environments. The sensor data is fused in EKF. Davison et al. proposed a new EKF based monocular vSLAM algorithm [17] utilizing computer vision techniques. The algorithm is real-time and uses a single camera that can recover the 3D trajectory when moving rapidly through a previously unknown scene.

As it is shown in the given applications of the EKF based SLAM algorithm, it is being widely used. Consistent maps are constructed and robust localization data are obtained. A variety of sensors are fused in EKF and successful results are obtained. The reason why these successful results are obtained is that it provides the optimal Minimum Mean-Square Error (MMSE) estimates for the pose and feature states, and the covariance matrix converges strongly [18].

On the other hand, in the applications that require the estimation of large-scale maps, EKF is relatively slow, because every single measurement generally affects all parameters of the Gaussian, therefore in the environments that have many landmarks, the update process takes very long time, computational complexity increases and consequently computer resources is not sufficient to update the map in real-time. The Compressed Extended Kalman Filter (CEKF) [5] decreases the computational load significantly without influencing the accuracy of the results. CEKF stores and maintains all of the obtained data in a local area with a cost proportional to the square of feature numbers in the environment. This data is transferred to the remaining part

of the global map with a cost similar to the SLAM [18].

Moreover, EKF linearizes all of the functions that the algorithm concerns about their current estimate points. This approximation gives rise to the errors, especially in the case of highly non linear functions. To prevent these kind of flaws, the problem is addressed by Unscented Kalman filter (UKF).

UKF was first published in 1997 by Julier and Uhlmann [19]. In the UKF the state distribution is again represented by a Gaussian Random Variable (GRV), but is now specified using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the GRV, and when propagated through the true non-linear system, captures the posterior mean and covariance precisely up to the 3^{rd} order for any nonlinearity. In order to do that, the unscented transform (UT) is used [18]. By sampling a Gaussian distribution with a fixed number of so called sigma-points, and passing these sigma-points through the desired nonlinear function or transformation, the UT avoids linearisation by taking explicit derivatives (Jacobians), which can be very hard in some cases. Once the sigma-points are passed through the nonlinear function, mean and covariance of the resulting transformed distribution can be retrieved from them. The UT sampling is a deterministic sampling, in contrast to techniques like particle filters, that sample randomly. This way, the number of samples can be kept small, compared to particle filters: To sample an n-dimensional distribution, $2n+1$ sigma-points are necessary. Further improvements on the UT, like [20] reduce this number to $n + 2$ [21].

2.1.3 Particle Filter Based SLAM

Rao-Blackwellized particle filter solution to the SLAM problem was introduced first in [7] by Montemerlo et al. and it is also known as FastSLAM. The algorithm utilizes the fact that estimated landmark positions are conditionally independent from the trajectory of the mobile robot. FastSLAM directly represents the nonlinear process model and non-Gaussian distribution unlike the EKF based algorithm which linearizes the process and measurement models using first order Taylor series expansion. Application of particle filter directly to the SLAM problem is not feasible if the state-space dimension is reasonably high. Utilizing Rao-Blackwellization (R-B) joint state is partitioned according to the product rule $P(\mathbf{x}_1, \mathbf{x}_2) = P(\mathbf{x}_2 \mid \mathbf{x}_1)P(\mathbf{x}_1)$ and consequently sample space is reduced. Also $P(\mathbf{x}_2 \mid \mathbf{x}_1)$ must be represented analytically, and then only $P(\mathbf{x}_1)$ is sampled such that $\mathbf{x}_1^{(i)} \approx P(\mathbf{x}_1)$. The joint distribution is represented by the set $(\mathbf{x}_1^{(i)}, P(\mathbf{x}_2 \mid \mathbf{x}_1^{(i)})_i^N$ and marginal statistics

$$P(\mathbf{x}_2) \approx \frac{1}{N} \sum_i^N P(\mathbf{x}_2 \mid \mathbf{x}_1^{(i)}) \tag{2.5}$$

can be obtained with greater accuracy than is possible by sampling over the joint space [1]. The joint SLAM state is shown as the multiplication of the vehicle trajectory and map component:

$$P(\mathbf{X}_{0:k}, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) = P(\mathbf{m} \mid \mathbf{X}_{0:k}, \mathbf{Z}_{0:k})P(\mathbf{X}_{0:k} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \tag{2.6}$$

In Eq. (2.6) the probability distribution involves all of the states of the vehicles up and including time instant k because of the fact conditioning on

the trajectory makes the map landmarks independent and this is the distinctive feature of FastSLAM and the reason for its speed [1]. The map is represented as a set of independent Gaussians which can be processed with linear rather than quadratic complexity [1]. In this form FastSLAM is principally a Rao-Blackwellised state, where the trajectory is indicated by weighted samples and the map is calculated analytically. Likewise, a set of particle weights, trajectory hypotheses and associated map hypotheses $(\mathbf{w}_k^{(i)}, \mathbf{X}_{0:k}^{(i)}, P(\mathbf{m} \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k}))_i^N$ are used to represent the joint distribution at time k. The maps are composed of a set of independent Gaussian distributions:

$$P(\mathbf{m} \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k}) = \prod_j^M P(\mathbf{m}_j \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k}) \tag{2.7}$$

Pose estimation of the mobile robot is carried out via particle filtering and the map of the environment is built using EKF. Interested readers may refer to [1] for more information.

2.1.4 Appearance Based SLAM

Filter based solutions proposed for large-scale SLAM problems are incapable for handling challenging phenomena such as loop closure and data association. Appearance based localization and mapping techniques have recently received significant attention and have been developed to utilize the rich appearance information acquired by visual sensors. They do not rely on position priors and are a useful approach to loop closure detection, since appearance based techniques can perform a global search through previously seen locations [22]. Environment is modeled exploiting well known SIFT

or SURF features, Discrete Cosine Transform (DCT), multidimensional histograms and Fourier Transforms. Extracted features in all of the models are matched via L1 (Manhattan Distance) or L2 (Euclidian Distance) [23].

Cummins and Newman proposed the Fast Appearance Based Mapping (FAB-MAP) method to map the large scale environments and to detect loop closures [24]. They extracted SURF visual features in the large areas and utilized the bag-of-words approach to generate a vocabulary. Chow-Liu tree is used to calculate the co-occurence statistics of the visual words that are in the vocabulary. To represent the co-occurence statistics a mutual information graph is constructed. The nodes describe a visual word and the links between nodes indicate weight (mutual information) in the graph. Confusion matrix is used to indicate the loop closure detections. The elements of the matrix show that a visited place is either a new location or detection of a loop closure. For very large scale navigation Cummins and Newman propose the second version of FAB-MAP which is a new formulation of appearance only SLAM [25]. The proposed system is highly scalable. The scalability of the system is achieved by defining a sparse approximation to the FAB-MAP model suitable for implementation using an inverted index. Lui and Jarvis presented a pure vision based topologic SLAM technique that uses appearance based place recognition system [9]. They use a mobile robot which estimates its motion via visual odometry and recognise places while performing concurrent localization and mapping.

2.1.5 Map Representation

Constructed maps are represented in different ways depending on the task that a mobile robot performs. Particularly there are four different kinds of

map representation:

Metric Map

In metric map representation, special features extracted from the environment using various sensors such as laser, sonar, inertial or visual are used. These are the features such as lines, corners, curves or planes that have geometrical representations. Metric map representation is used frequently in filter based SLAM algorithms and provides direct information about free collision trajectories for the navigation of a mobile robot [23].

Topologic Map In the environments where metric map representation cannot be used efficiently, topologic map representation is utilized. Topologic map is extremely suitable for the appearance based SLAM algorithms that are improved for large scale environments. It accelerates the navigation process of a mobile robot providing decrease in the computer memory consumption. Graph structures are used in topologic map representation. All nodes indicate the appearance data of the locations and links between nodes represent traversable paths between locations. There is no geometric relationship in the topologic map. Cummins and Newmans work in [24] is a good example to show the efficiency of the topologic map.

Hybrid Map This kind of representation incorporates the high performances of the metric and topologic maps. Topologic maps represent large scale environments in a compact form. Metric maps deal with the uncertainties of the pose and landmark states. Metric map indicates the spatial relationship between the topologic maps elements.

Occupancy Grid Occupancy Grid Mapping addresses the map generation problem from uncertain and noisy measurement data. This mapping assumes that the pose of the mobile robot is known. In occupancy grid the

map of the environment is represented as an evenly spaced field of binary random variables. The random variables show the presence of an obstacle at that location in the environment.

2.2 Feature Extraction and Matching

Passive vision based sensors are used to extract, match and track the visual features that are artificial or natural in the environment. Also 3D location of the features and the pose of the mobile robot are estimated utilizing computer vision techniques. Because of the fact that we assume the visual sensor is attached to the body of a mobile robot, computing the pose of the camera will give rise the calculation of the robot pose. Hence, the goal is to track some keypoints precisely. In this subsection, we present how the extraction of features is performed and how the keypoints are matched, in order to compute a rigid transformation between a couple of frames. To find a set of corresponding locations in different images, generally between two consecutive images, point features such as Harris corners, SIFT or SURF can be used. There are two main approaches to find feature points and their correspondences [26]. In the first approach, features are found in a single image and then tracked accurately using a local search technique such as correlation or least squares. This method is suitable when images are taken rapidly or from nearby viewpoints. The second approach detects the features in all frames of a video sequence and then match them based on their local appearance and it is more suitable when a large amount of motion is applied.

2.2.1 Feature Detectors

Tracking and matching the visual features in an accurate way depends on the quality of the extracted features. To show the reliability of the extracted features three sample patches are shown in Fig. 2.4. A textureless patch is indistinguishable and cannot be recognized easily. To localize a patch easily it should have large contrast changes (gradients), but straight line segments at a single orientation suffer from the aperture problem [27], [28], [29], i.e., it is only possible to align the patches along the direction normal to the edge direction. Patches that have gradients at least two different orientations are the easiest to localize.

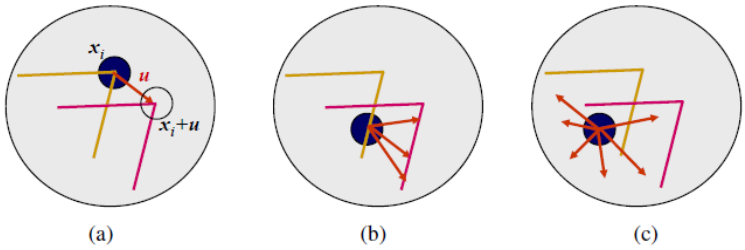


Figure 2.4: Different Orientations
[26]

In a comparison of two different image patches, sum of squared difference estimator can be used to formalize the intuition which is underlying feature detection [26],

$$E_{WSDD}(\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2 \tag{2.8}$$

where I_0 and I_1 are the two images being compared, $\mathbf{u} = (u, v)$ is the displacement vector, $w(\mathbf{x})$ is a spatially varying weighting function, and the summation i is over all the pixels in the patch. When performing feature

detection, we do not know which other image locations the feature will end up being matched against. Therefore, we can only compute how stable this metric is with respect to small variations in position $\triangle\mathbf{u}$ by comparing an image patch against itself, which is known as an auto-correlation function:

$$E_{AC}(\triangle\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \triangle\mathbf{u}) - I_0(\mathbf{x}_i)]^2 \tag{2.9}$$

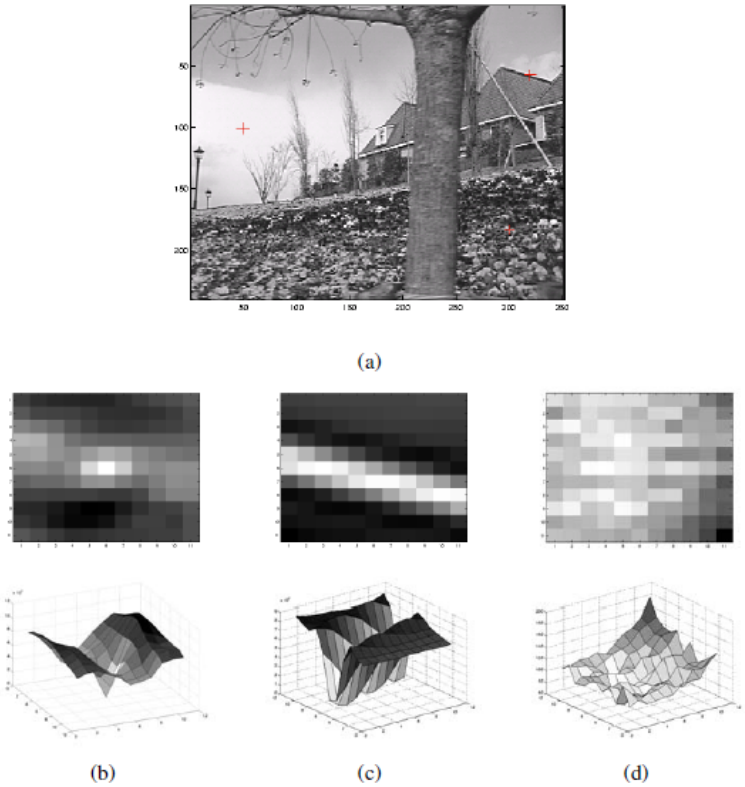


Figure 2.5: Three auto-correlation surfaces
[26]

Note how the auto-correlation surface for the textured flower bed (Figure 2.5 b) and the red cross in the lower right quadrant of (Figure 2.5 a) exhibits a strong minimum, indicating that it can be well localized. The

correlation surface corresponding to the roof edge (Figure 2.5 c) has a strong ambiguity along one direction, while the correlation surface corresponding to the cloud region (Figure 2.5 d) has no stable minimum.

Using a Taylor Series expansion of the image function $I_0(\mathbf{x}_i + \triangle\mathbf{u}) \approx I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i)\triangle\mathbf{u}$ we can approximate the auto-correlation surface as,

$$E_{AC}(\triangle\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \triangle\mathbf{u}) - I_0(\mathbf{x}_i)]^2 \tag{2.10}$$

$$\approx \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i)\triangle\mathbf{u} - I_0(\mathbf{x}_i)]^2 \tag{2.11}$$

$$= \sum_i w(\mathbf{x}_i)[\nabla I_0(\mathbf{x}_i)\triangle\mathbf{u}]^2 \tag{2.12}$$

$$= \triangle\mathbf{u}^T\mathbf{A}\triangle\mathbf{u} \tag{2.13}$$

$$\tag{2.14}$$

where

$$\nabla I_0(\mathbf{x}_i) = (\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y})(\mathbf{x}_i) \tag{2.15}$$

is the image gradient at $\mathbf{x}_i$. This gradient can be computed using a variety of techniques [30]. The classic Harris detector [31] uses a [-2 -1 0 1 2] filter, but more modern variants [30], [32] convolve the image with horizontal and vertical derivatives of a Gaussian.

The auto-correlation matrix $\mathbf{A}$ can be written:

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{2.16}$$

where we have replaced the weighted summations with discrete convolutions

with the weighting kernel w. This matrix can be interpreted as a tensor (multiband) image, where the outer products of the gradients ∇I are convolved with a weighting function w to provide a per-pixel estimate of the local (quadratic) shape of the auto-correlation function.

As first shown by [33], the inverse of the matrix $\mathbf{A}$ provides a lower bound on the uncertainty in the location of a matching patch. It is therefore a useful indicator of which patches can be reliably matched. The easiest way to visualize and reason about this uncertainty is to perform an eigenvalue analysis of the auto-correlation matrix $\mathbf{A}$, which produces two eigenvalues (λ_0 , λ_1) and two eigenvector directions. Since the larger uncertainty depends on the smaller eigenvalue, i.e., $\lambda_0^{-1/2}$, it makes sense to find maxima in the smaller eigenvalue to locate good features to track [34].

Harris Corners: Harris corner point detector was proposed in 1988 by Harris and Stephens [31]. Harris also showed its value for efficient motion tracking and 3D structure from motion recovery, and the Harris corner detector has been widely used for many other image matching tasks. Despite the fact that these feature detectors are usually called corner detectors, they are not selecting just corners, but rather any image location that has large gradients in all directions at a predetermined scale [35]. The formulation of the scale is proposed by Harris in the following form,

$$\mathbf{R} = det(\mathbf{A}) - \alpha trace(\mathbf{A}) \tag{2.17}$$

$$= \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2 \tag{2.18}$$

The windows that have a score R greater than a certain value are extracted as corners. They are good tracking points.

Scale Invariant Feature Transform: The Scale Invariant Feature Transform (SIFT) is a method developed by David Lowe [35] and intensely used in vision and robotics applications. SIFT method extracts features from images that are invariant to scale, rotation, illumination and viewpoint and allows to perform tasks such as object detection and recognition, computing geometrical transformations between images. It has 4 major stages to generate the set of image features:

1) Scale-space extrema detection: This process searches over all scales and image locations. In this stage a difference-of-Gaussian function is implemented to identify potential interest points that are invariant to scale and orientation. When a mobile robot moves in an environment, the features are seen both larger and smaller regarding the vantage point of the visual sensor mounted to the vehicle. For the vSLAM problem providing the scale invariance condition for the extracted features ensures consistent map building and localization for a mobile robot. Hence, it is a fundamental requirement. The theory of the scale space is based on Lindeberg' s work in [36] and the main idea is shown in (Figure 2.6). The scale space is defined as the convolution of an image I with a Gaussian G,

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.19}$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{2.20}$$

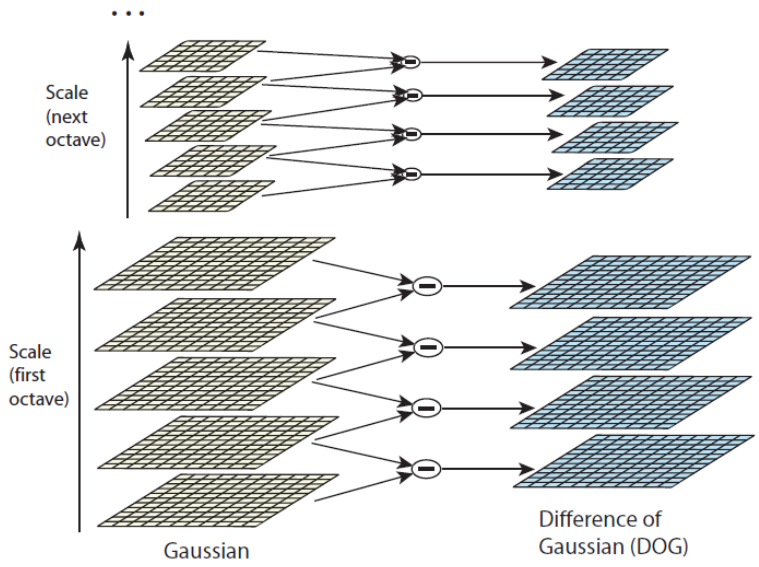


Figure 2.6: Difference of Gaussians at different scales [35]

The DoG is the difference between two layers in scale space along the σ axis:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{2.21}$$

$$= L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.22}$$

This provides a close approximation to the scale-normalized Laplacian of Gaussian $\sigma^2\nabla^2 G$, as shown by Lindeberg [36]:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{2.23}$$

and thus,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2\nabla^2 G \tag{2.24}$$

The σ^2 defines the invariance for the scale. At the end of the stage the

keypoint candidates are found via extrema in DoG that approximates the Laplacian $\sigma^2\nabla^2 G$, however this process finds the unstable keypoints that have low contrast and are poorly localized along edges. So, these unstable keypoints must be rejected.

2) Keypoint localization: To reject the keypoints that have low contrast and sensitive to noise or localized poorly along an edge, a detailed fit is required to the data around the keypoint for location, scale, and ratio of principal curvatures. This process is achieved by the method which Brown developed. In this method, a 3D quadratic function is fitted to the local sample points to determine the interpolated location of the maximum which uses the Taylor expansion up to the quadratic terms of the scale-space function $D(x, y, \sigma)$:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.25}$$

where D and its derivatives are evaluated at the sample point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, $\bar{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{x}$ and setting it to zero, giving

$$\bar{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{(-1)}\frac{\partial D}{\partial \mathbf{x}} \tag{2.26}$$

The function value at the extremum, $D(\bar{\mathbf{x}})$, is useful for rejecting unstable extrema with low contrast. This can be obtained:

$$D(\bar{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\bar{\mathbf{x}} \tag{2.27}$$

Assigning a threshold value to $|D(\bar{\mathbf{x}})|$ results in the feature points extraction that are stable. From the point of keypoints poorly located along an edge, principal curvatures are computed using a Hessian matrix at the location and scale of the keypoints:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.28}$$

The reason why principal curvature are being calculated is that the difference-of-Gaussian function has a strong response along edges, even if the location along the edge is poorly determined and therefore unstable to small amounts of noise. A poorly defined peak in the difference-of-Gaussian function will have a large principal curvature across the edge but a small one in the perpendicular direction [35]. The principal curvatures of D are proportional to the eigenvalues of $\mathbf{H}$. Herein, to check the ratio of principal curvatures, the following ratio value is used:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{2.29}$$

where r is the ratio of largest and smaller eigenvalues of $\mathbf{H}$.

Thus far, we determined stable keypoint candidates. In the next step we assign an orientation to these keypoints which will be used to define keypoint descriptors.

3) Orientation assignment: The magnitude and orientation is calculated for all pixels around the keypoint.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.30}$$

$$\theta(x,y) = tan^{-1}(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}) \tag{2.31}$$

An orientation histogram is formed from the gradient orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian weighted circular window. Peaks in the orientation histogram correspond to dominant directions of local gradients [35].

4) Keypoint descriptor: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the leftmost of Figure 2.7. These are weighted by a Gaussian window. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples.

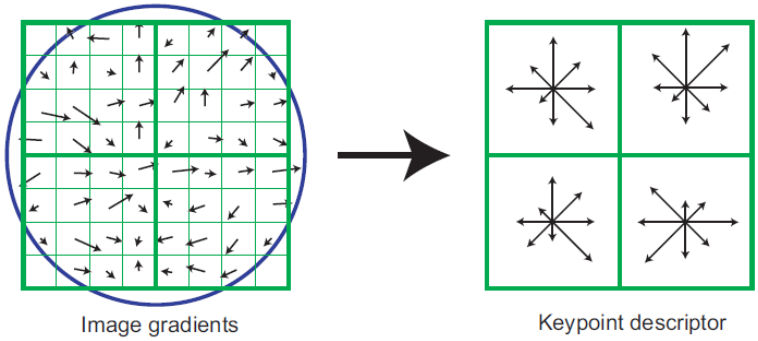


Figure 2.7: Image gradients and keypoint descriptor [35]

Speeded Up Robust Feature: The Speeded Up Robust Feature (SURF) is a robust detector and descriptor that is presented by Bay in [37]. This is

a method to extract feature points, that readily can be matched between images to detect and recognize, to compute geometrical transformations between images and to use in structure from motion method. The main difference from SIFT features is the performance, providing low computational complexity through an efficient use of integral images for the image convolutions, Hessian matrix-based detector and sums of approximated 2D Haar wavelet responses for the descriptor. The standard version of SURF is nearly 4 times faster than SIFT.

2.2.2 Feature Matching

Once we have extracted features and their descriptors from two or more images, the next step is to find matched features between images. In order to do so, we assume that there is enough overlapped area that have the same features between two images. To find potential matches we calculate the Euclidean distance between the feature descriptors through a nearest neighbour search. Given a Euclidean distance metric, the simplest matching strategy is to set a threshold (maximum distance) and to return all matches from other images within this threshold. Setting the threshold too high results in too many false positives, i.e., incorrect matches being returned. Setting the threshold too low results in too many false negatives, i.e., too many correct matches being missed. This matching method is widely used when images are taken from nearby viewpoints or in rapid succession. When a large amount of motion or appearance change is expected *detect then track* method is much more suitable for matching. In this matching method a set of feature locations are found in the first image and then searched for their corresponding locations in subsequent images. In the latter process good

visual features must be selected to track. Regions containing high gradients in both directions provide stable locations at which to find correspondences. However, if the illumination change is large in the images, explicitly compensating for such variations or using normalized cross-correlation may be preferable. If the search range is large, it is also often more efficient to use a hierarchical search strategy, which uses matches in lower-resolution images to provide better initial guesses and hence speed up the search. Alternatives to this strategy involve learning what the appearance of the patch being tracked should be and then searching for it in the vicinity of its predicted position [26].

Over longer image sequences, the appearance of the features being tracked can undergo larger changes. An affine motion model is proposed as a feasible solution that compares the original patch to later image locations. Shi and Tomasi (1994) first compare patches in neighbouring frames using a translational model and then use the location estimates produced by this step to initialize an affine registration between the patch in the current frame and the base frame where a feature was first detected. In their system, features are only detected infrequently, i.e., only in regions where tracking has failed. In the usual case, an area around the current predicted location of the feature is searched with an incremental registration algorithm. This tracking process is called the Kanade Lucas Tomasi (KLT) tracker [26].

Chapter III

3 Performance Improvement in vSLAM Using Stabilized Feature Points

In this chapter, we propose a performance improvement technique for vSLAM that extracts stabilized Harris corner features using template matching based stabilized video sequences. When a non-holonomic wheeled mobile robot (WMR) navigates in an unknown environment, some undesired phenomena such as vibrations on the mobile robot and the speed bump constructions in the environment might occur. With the proposed technique, these problems are eliminated, and as a result stabilized feature extraction is achieved. Stabilized keypoint extraction ensures both consistency in map building and localization of the mobile robot.

3.1 Sensor Fusion Architecture

The sensor fusion architecture developed in this work is shown in Figure 3.1 and composed of several modules. Data generated by both the camera and the odometry are used in feature extraction (FE) and dead reckoning (DR) blocks, respectively. The output of FE is the observation, and the output of DR is the robot state prediction. In measurement prediction block, predicted states obtained from the robot model are used and the sensor measurement

model is utilized to predict the measurements. In matching module, measurement predictions are subtracted from observations to calculate the innovation and innovation covariance. The output of the matching block is transferred to EKF update block to estimate the non-holonomic WMR states and build the map.

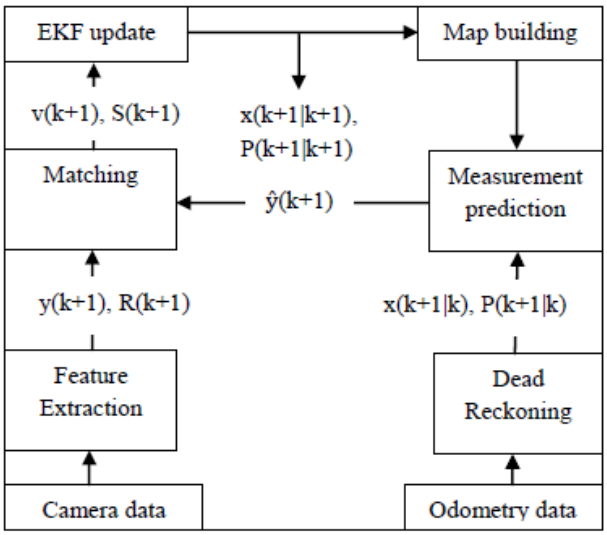


Figure 3.1: Sensor fusion architecture

3.2 Mathematical Model of the Mobile Robot

The non-holonomic WMR shown in Figure 3.2 includes two driving wheels and a back caster that are non deforming. The robot moves on the horizontal plane and the contact of the wheels with the ground is assumed to satisfy rolling without any skidding or slipping.

3.2.1 Kinematic Model

In the kinematic modelling of the non-holonomic WMR, orientation must be considered since it affects the robot movement along x and y directions based on the kinematic constraints of the system.

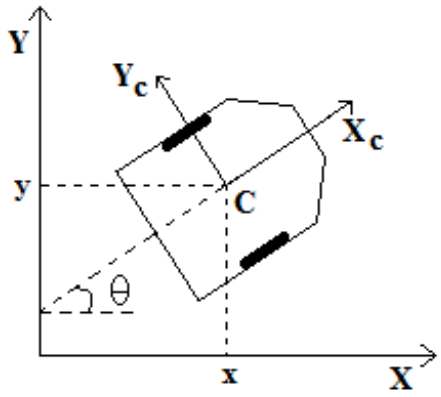


Figure 3.2: Non-holonomic wheeled mobile robot

The kinematic model of the NWMR is described by the following equations [38]:

$$\dot{x} = vcos(\theta) \tag{3.1}$$

$$\dot{y} = vsin(\theta) \tag{3.2}$$

$$\dot{\theta} = w \tag{3.3}$$

or, can be written in a more compact form as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{3.4}$$

where $\mathbf{x} = [x, y, \theta]^T$ is the pose (position and orientation) of the centre of

mass of the mobile robot C, with respect to world coordinate frame O, $\mathbf{u} = [v, w]^T$ is the control input vector, where v is the linear velocity and w is the angular velocity of the mobile robot, respectively. Using Eulers forward difference approximation for the derivative, the discrete form of the mobile robot kinematic model can be written as:

$$x_{k+1} = x_k + Tvcos(\theta_k) \tag{3.5}$$

$$y_{k+1} = y_k + Tvsin(\theta_k) \tag{3.6}$$

$$\theta_{k+1} = \theta_k + wT \tag{3.7}$$

or in a more compact form as

$$\mathbf{x_{k+1}} = f(\mathbf{x_k}, \mathbf{u_k}) \tag{3.8}$$

$$f(\mathbf{x_k}, \mathbf{u_k}) = \begin{bmatrix} f_x \\ f_y \\ f_\theta \end{bmatrix} = \begin{bmatrix} x_k + Tvcos(\theta_k) \\ y_k + Tvsin(\theta_k) \\ \theta_k + wT \end{bmatrix} \tag{3.9}$$

where k is the discrete time index, T is the sampling period and $f(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is a nonlinear mapping [39]. In order to implement EKF, this nonlinear system must be linearized. In [40], it is shown that applying the Taylor series approximation to the right-hand side of Eq. (3.4) and ignoring the higher order terms yields the following linear state-space model of the mobile robot:

$$\mathbf{x}(k+1) = A(k)\mathbf{x}(k) + B(k)\mathbf{u}(k) \tag{3.10}$$

The state $A(k)$ and input $B(k)$ matrices are defined as follows:

$$A(k) = \begin{bmatrix} \frac{\partial f_x}{\partial x_k} & \frac{\partial f_x}{\partial y_k} & \frac{\partial f_x}{\partial \theta_k} \\ \frac{\partial f_y}{\partial x_k} & \frac{\partial f_y}{\partial y_k} & \frac{\partial f_y}{\partial \theta_k} \\ \frac{\partial f_\theta}{\partial x_k} & \frac{\partial f_\theta}{\partial y_k} & \frac{\partial f_\theta}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -Tv_k sin(\theta_k) \\ 0 & 1 & Tv_k cos(\theta_k) \\ 0 & 0 & T \end{bmatrix} \tag{3.11}$$

$$B(k) = \begin{bmatrix} \frac{\partial f_x}{\partial u_k} & \frac{\partial f_x}{\partial w_k} \\ \frac{\partial f_y}{\partial u_k} & \frac{\partial f_y}{\partial w_k} \\ \frac{\partial f_\theta}{\partial u_k} & \frac{\partial f_\theta}{\partial w_k} \end{bmatrix} = \begin{bmatrix} Tcos(\theta_k) & 0 \\ Tsin(\theta_k) & 0 \\ 0 & T \end{bmatrix} \tag{3.12}$$

3.2.2 Camera Sensor Model

Ideal pin hole camera model is used as a measurement model. Acquired measurements from the camera generate the measurement vector $\mathbf{y}$,

$$\mathbf{y} = [y_{1k}, y_{2k}, ..., y_{pk}]^T \tag{3.13}$$

where p is the number of the features observed at a particular time index k. At the same time, all the observed image features build up the map of the environment. At any time k, for one observed image feature camera model implies:

$$\begin{bmatrix} m_{ix} \\ m_{iy} \end{bmatrix} = \begin{bmatrix} O_x + f_c \frac{s^c_{ix}}{s^c_{iz}} \\ O_y + f_c \frac{s^c_{iy}}{s^c_{iz}} \end{bmatrix} \text{ for } i\text{= 1, 2, 3, ..., } p \tag{3.14}$$

where f_c is the focal length of the camera, (O_x, O_y) is the principal point of the image plane in pixels, $s^c = [s^c_{ix}, s^c_{iy}, s^c_{iz}]^T$ is the 3D location of the extracted feature with respect to the camera frame. 3D location of the i^{th}

feature with respect to the world coordinate frame is given as [41]:

$$q_i = [X_i, Y_i, Z_i]^T = r + R_C^W s_i^c \tag{3.15}$$

where q_i is the 3D location of the image feature in world frame, R_C^W is the rotation matrix that defines the orientation of the camera frame with respect to the world frame, r is the 3D translation vector from world frame to camera frame. A rotation matrix can be parameterized by three independent variables such as Euler angles. Due to the planar robot motion assumption, the orientation matrix will be just in terms of the yaw angle [42]:

$$R_C^W = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

In Eq. (3.16), θ (heading angle) is taken from the estimated states of the EKF that will be summarized in the next section. By rearranging Eq. 3.15, one can calculate the s_i^c as:

$$s_i^c = R_W^C(q_i - r) \tag{3.17}$$

where R_W^C is simply the transpose of the rotation matrix R_C^W. Plugging Eq. (3.17) into the measurement model yields the extracted feature location in image plane:

$$\begin{bmatrix} m_{ix} \\ m_{iy} \end{bmatrix} = \begin{bmatrix} O_x + f_c \frac{cos(\theta)(X_i - r_x) + sin(\theta)(Y_i - r_y))}{Z_i - r_z} \\ O_y + f_c \frac{-sin(\theta)(X_i - r_x) + cos(\theta)(Y_i - r_y)}{Z_i - r_z} \end{bmatrix} \tag{3.18}$$

The measurement Jacobian H_k is calculated by taking the derivative of the right hand side of the Eq. (3.18) with respect to the states of the mobile robot $\mathbf{x_k}$. Thus,

$$\mathbf{H(k)} = \begin{bmatrix} \frac{\partial m_{ix}}{\partial r_x} & \frac{\partial m_{ix}}{\partial r_y} & \frac{\partial m_{ix}}{\partial \theta} & \frac{\partial m_{ix}}{\partial X_i} & \frac{\partial m_{ix}}{\partial Y_i} \\ \frac{\partial m_{iy}}{\partial r_x} & \frac{\partial m_{iy}}{\partial r_y} & \frac{\partial m_{iy}}{\partial \theta} & \frac{\partial m_{iy}}{\partial X_i} & \frac{\partial m_{iy}}{\partial Y_i} \end{bmatrix} \tag{3.19}$$

$$\mathbf{H(k)} = \frac{f_c}{Z_i - r_z} \begin{bmatrix} -cos(\theta) & -sin(\theta) & -sin(\theta)(X_i - r_x) + cos(\theta)(Y_i - r_y) & cos(\theta) & sin(\theta) \\ sin(\theta) & -cos(\theta) & -cos(\theta)(X_i - r_x) - sin(\theta)(Y_i - r_y) & -sin(\theta) & cos(\theta) \end{bmatrix} \tag{3.20}$$

Observation and measurement prediction data are fused in EKF to calculate the innovation and innovation covariance.

3.3 Extended Kalman Filter

The mobile robot navigates in an unknown environment, without any a priori knowledge about the map, takes measurements to extract feature points and consequently localizes itself. External (camera) and internal (odometry) sensory data will be fused in EKF. The robot pose $\mathbf{x}$ and the locations of the extracted feature points $\mathbf{X_F}$ with respect to the world frame can be stacked in a new state vector as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x} \\ \mathbf{X_F} \end{bmatrix} \tag{3.21}$$

where $\mathbf{x} = [x, y, \theta]^T$ defines position and orientation of the robot, and is governed by the following nonlinear model:

$$\mathbf{x_{k+1}} = f(\mathbf{x_k}, \mathbf{u_{k+1}}, \eta_k) \tag{3.22}$$

$$\mathbf{y_{k+1}} = h(\mathbf{X_{k+1}}, \xi_k) \tag{3.23}$$

where η_k and ξ_k are the process and the measurement noise, which are modeled as zero-mean, independent Gaussian distributions with covariance matrices F_k and G_k, respectively.

The second element of $\mathbf{X}$ is defined as:

$$\mathbf{X_F} = \begin{bmatrix} X_{fi} \\ Y_{fi} \end{bmatrix} \text{ for } i\text{= 1, 2, ..., } n \tag{3.24}$$

where $\mathbf{X_F} = [X_{fi}, Y_{fi}]^T$ are the locations of the extracted features with respect to the world frame and added to the map at time k. Since the positions of the extracted features are not changed, they remain at the same locations during the navigation i.e.;

$$\mathbf{X_{F,k+1}} = \begin{bmatrix} X_{fi} \\ Y_{fi} \end{bmatrix}_{k+1} = \mathbf{X_{F,k}} \tag{3.25}$$

Linearisation of Eqs. (3.22) and (3.25) with respect to X imply new Jacobians for the process model [38]:

$$\bar{A} = \begin{bmatrix} A & O_1 \\ O_1^T & I \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ O_2 \end{bmatrix} \tag{3.26}$$

$$\bar{H} = \begin{bmatrix} \frac{\partial m_{ix}}{\partial(r_x, r_y, \theta, X_{fi}, Y_{fi}{}_{i=1,2,\ldots,n})} \\ \frac{\partial m_{iy}}{\partial(r_x, r_y, \theta, X_{fi}, Y_{fi}{}_{i=1,2,\ldots,n})} \end{bmatrix} \tag{3.27}$$

where $A \in R^{(3\times3)}$, $O_1 \in R^{(3\times2n)}$ (zero matrix), $I \in R^{(2n\times2n)}$ (identity matrix), $B \in R^{(3\times2)}$ and $O_2 \in R^{(2n\times2)}$ (zero matrix) with n being the number of features extracted at time k. With this framework, the following algorithm summarizes the recursions involved in computing the EKF [43]:

$$\mathbf{X_{k+1|k}} = f(\mathbf{X_k}, \mathbf{u_{k+1}}) \tag{3.28}$$

$$P_{k+1|k} = \bar{A}_{k+1,k} P_k \bar{A}_{k+1,k}^T + \bar{F}_k \tag{3.29}$$

$$K_{k+1} = P_{k+1|k} \bar{H}_{k+1}^T [\bar{H}_{k+1} P_{k+1|k} \bar{H}_{k+1|k}^T + G_k]^{-1} \tag{3.30}$$

$$\mathbf{X_{k+1}} = \mathbf{X_{k+1|k}} + K_{k+1}(\mathbf{y_{k+1}} - h(\mathbf{X_{k+1|k}})) \tag{3.31}$$

$$P_{k+1} = (I - K_{k+1} \bar{H}_{k+1}) P_{k+1|k} \tag{3.32}$$

where $\bar{F}_k$ is the covariance matrix of the combined state $\mathbf{X}$. To initialize the filter, $\mathbf{X_0}$ and P_0 are set to some arbitrary random values.

3.4 Stabilized Feature Point Extraction

Extracting feature points accurately increases the performance of vSLAM algorithm since they are used in EKF measurement update. It provides

improvement in both map building and localization of the mobile robot.

Video stabilization is one of the most crucial video processes that reduces the blurring level of image sequences and unwanted camera motions. Extracting point features from stabilized video frames improves the consistency of the static landmarks and provides robust matching between corresponding points. Proposed video stabilization method in this work is based on a template matching that uses the sum of absolute differences (SAD) algorithm:

$$SAD = \sum_{(i,j)\in W} |I_1(i,j) - I_2(x+i, y+j)| \tag{3.33}$$

where I_1 and I_2 are two consecutive image frames. $I_1(i,j)$ and $I_2(x+i, y+j)$ defines the pixel intensity values. In I_1, a window W, e.g. size of (15 x 15), is generated around an interest point. Meanwhile, each pixel in the second video frame is scanned by shifting this window along horizontal (x) and vertical (y) directions. Note that the intensity values in the second window is subtracted from those values in the first window. The absolute values of all these pixel intensities in W are summed. If there is a correct match, the SAD function gives a near 0 value. Thus, a similar window is created in the second video frame [44]. Scan process can be applied both over the entire image or just using a region of interest. In each subsequent video frame, SAD algorithm determines the camera motion relative to the previous frame. It uses this information to remove unwanted translational camera motions and generate a stabilized video.

Feature extraction from consecutive images is one of the essential steps of vSLAM applications. In this work, extracted image features are corners that are obtained via Harris corner detector. Some example images and

extracted Harris corner features are shown in Figure 3.3. A video sequence is deliberately subject to jitter and noise in Figure 3.3 (a) and extracted Harris corner features from this image are shown in Figure 3.3 (b). It is then stabilized using the proposed technique and the resultant image is depicted in Figure 3.3 (c). Extracted features are shown in Figure 3.3 (d) where there is an increase in the number of consistent features due to video stabilization.

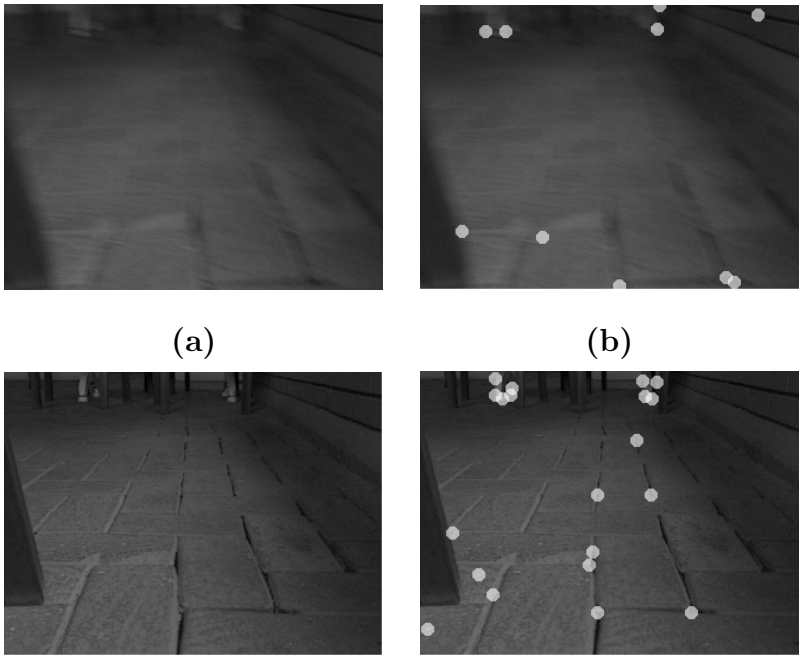


Figure 3.3: Stabilized feature point extraction: (a) a sample image before video stabilization, (b) extracted Harris corner features before video stabilization, (c) a sample image after video stabilization, (d) extracted Harris corner features after video stabilization

Chapter IV

4 Under Vehicle Perception Using a Catadioptric Camera System

In recent years, under vehicle surveillance and the classification of the vehicles become an indispensable task that must be achieved for security measures in certain areas such as shopping centers, government buildings, army camps etc. The main challenge to achieve this task is to monitor the under frames of the means of transportations. In this chapter, we present a novel solution to achieve this aim. Our solution consists of three main parts: monitoring, detection and classification. In the first part we design a new catadioptric camera system in which the perspective camera points downwards to the catadioptric mirror mounted to the body of a mobile robot. Thanks to the catadioptric mirror the scenes against the camera optical axis direction can be viewed. In the second part we use speeded up robust features (SURF) in an object recognition algorithm. Fast appearance based mapping algorithm (FAB-MAP) is exploited for the classification of the means of transportations in the third part.

Since the conventional camera systems have limited field of view, realization of above task becomes infeasible. In such a scenario, conventional systems require many cameras that give rise to high computational cost. Moreover, displaying the under frames of the vehicles by typical perspective

cameras that have different orientations or a single rotating camera requires wide installation space and extensive calibration. On the other hand, because of the fact that catadioptric camera systems are able to capture omnidirectional images of the environments, i.e. providing 360 degree field of view, one can monitor the under frames of the vehicles, detect the undercovered materials and classify the vehicles just using a single catadioptric camera. This unique feature of the catadioptric cameras eliminates disadvantages of perspective cameras. Moreover, increase in the number of extracted features from panoramic images maintains stability for object detection and classification.

A catadioptric camera system consists of a convex mirror such as a parabolic, a spherical, an elliptical or a hyperbolic mirror and a single conventional perspective camera. They are also called as omnidirectional vision systems and have been studied extensively in [45, 46]. Catadioptric camera systems can be categorized into central and noncentral catadioptric systems. In a central catadioptric camera system convex mirror is aligned with a central camera where it has a single projection center. For more details, interested readers may refer to [47, 48]. Nevertheless, in practice, the real catadioptric cameras have to be treated as noncentral cameras since they have multiple effective viewpoints. Misalignment between the perspective lens and convex mirror, structural imperfection in the convex mirror types, inexact positioning of the perspective camera in one of the focal points of the convex mirror should cause the noncentrality [49]. Regarding the utilization of the multiple catadioptric cameras, different omnidirectional vision systems are designed for different tasks. Schönbein et al. propose two different catadioptric stereo camera systems in [50] that are the combination of

the catadioptric-perspective and catadioptric-catadioptric systems mounted on a car. In [51] Lui and Jarvis present vertically aligned stereo catadioptric system that has a variable vertical baseline. Gandhi and Trivedi design an omnidirectional stereo system for visualizing the nearby environment of a vehicle [52]. Schnbein et al. combine three catadioptric cameras and align them horizontally in [53] to increase the robustness of the ego motion estimation and localization by 3D features all around the autonomous vehicles.

From the point of under vehicle surveillance, various monitoring systems are proposed. In [54] a vehicle inspection system is proposed that uses an image mosaic generation technique for different perspective views. A mobile robot equipped with a 3D range sensor to inspect the under frames of the vehicles is offered by Sukumar et al. [55]. A combination of the vehicle recognition and the inspection system is proposed in [56] to improve safety precautions. In [57] an automatic under vehicle inspection system is utilized to monitor the under frames of the vehicles. Regarding the under vehicle surveillance in most of the proposed solutions, different computer vision and image processing algorithms are utilized with perspective cameras.

In this study we propose a new catadioptric camera system that consists of a perspective camera pointing downwards to the convex mirror mounted to the body of a mobile robot to monitor the under frames of the vehicles. We show how to solve one of the most common safety measure problems in structures where extra safety precautions must be taken by displaying under frames of the vehicles that cannot be dealt with conventional perspective cameras easily. While mobile robot navigates under the means of transportations, it starts to detect the hidden materials attached to the under vehicles and classify the vehicles utilizing the fast appearance based mapping (FAB-

MAP) algorithm [24]. If the robot detects a peculiar material such as a bomb it warns the detection of the material by drawing a line between the object image in the database and the object that is seen in the video frame.

4.1 Catadioptric Camera System

4.1.1 Catadioptric Camera Model

In the design of the catadioptric camera systems one important property that must be considered is determining the shapes of the mirrors in such a way that the single effective viewpoint condition is ensured. The reason why a single effective viewpoint is desirable is that it allows the derivation of the epipolar geometry of two omnidirectional images and it is a requirement for the generation of pure perspective images from the sensed images. Regarding our omnidirectional vision system, we used hyperbolic convex mirrors and the projection model that Mei et al. propose in [58]. In the following steps we summarize the imaging model (Figure 4.1):

1) The projective ray $\mathbf{x}$ coming from $\mathbf{X}$ intersects the unit spherical surface in M,

$$(M)_0 = \frac{\mathbf{X}}{\| \mathbf{X} \|} = (\frac{X}{\| \mathbf{X} \|}, \frac{Y}{\| \mathbf{X} \|}, \frac{Z}{\| \mathbf{X} \|})^T \tag{4.1}$$

where $\| \mathbf{X} \| = \sqrt{X^2 + Y^2 + Z^2}$.

2) Once the world points are projected onto the unit sphere, the points are changed to a new reference frame centered in $O_c = (0, 0, -\xi)$,

$$(M)_{0_c} = (\frac{X}{\| \mathbf{X} \|}, \frac{Y}{\| \mathbf{X} \|}, \frac{Z}{\| \mathbf{X} \|} + \xi)^T \tag{4.2}$$

where ξ mirror parameter is the distance between O_c and sphere center O.

3) These points are projected onto the normalized image plane $Z = \psi - 2\xi$. The intersection of the projective ray $\mathbf{y}$ with the plane is the catadioptric image of the 3D point $\mathbf{X}$,

$$\mathbf{x_i} = (\frac{X}{Z + \xi \parallel \mathbf{X} \parallel}, -\frac{Y}{Z + \xi \parallel \mathbf{X} \parallel}, 1)^T = f_i(\mathbf{X}) \tag{4.3}$$

4) The final projection matrix includes a camera projection matrix K with γ the generalized focal length, (u_0, v_0) the principal point, s the skew and r the aspect ratio [15].

$$\mathbf{p} = k(\mathbf{x_i}) = \begin{bmatrix} \gamma & \gamma s & u_0 \\ 0 & \gamma r & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x_i} = K\mathbf{x_i} \tag{4.4}$$

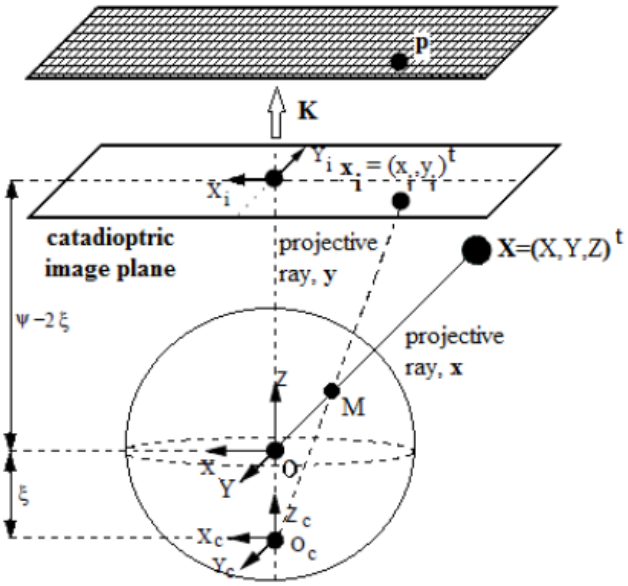


Figure 4.1: Modelling central catadioptric image formation

4.1.2 Catadioptric Camera System

The catadioptric camera system proposed in this paper is a combination of a hyperbolic mirror and a perspective camera. The hyperbolic mirror is attached to a plexiglass plate and it is passed through a four sided transparent plexiglass tube in such a way that the mirror is settled down in the base. Top side of the tube is covered with a hole centered transparent plate that the camera lens is able to point down to the hyperbolic mirror. Some example photos taken using the catadioptric system are depicted in Figure 4.2. Since the perspective camera points downwards we can see the ceiling of the laboratory in these images. Once we designed this system, we mounted it to the body of a non-holonomic mobile robot. The main advantage of such a system is to monitor the vehicle under frames that cannot be achieved easily using conventional camera systems. Other benefits obtained from this system can be listed as: It increases the field of view and as a result not only the frontal direction but also the right, left and back sides of the mobile robot are displayed. The number of extracted features from single catadioptric image is higher than a perspective image and so matching between two consecutive images taken from catadioptric cameras gives rise to much more consistent results in terms of object recognition and classification, localization and mapping. In this study we use just a single catadioptric camera for object recognition and vehicle classification that is able to monitor upper side of the camera mounting area. One can also design a catadioptric stereo system to utilize for 3D reconstruction, visual simultaneous localization and mapping, structure from motion and pose estimation etc.

Figure 4.2: Catadioptric images

4.2 Object Recognition

In a typical object recognition system, extracted features from a test object are matched against the features of the object model database to determine the identity of an object as shown in Figure 4.3. There are two main approaches in object recognition: model-based recognition and appearance-based recognition. In model-based recognition problem, an object model is being used and it is subjected to geometric transformation that maps the model in 3D world into the camera sensor coordinate frame. In such a recognition approach, efficient algorithms for estimating geometric transformations are central to many model-based recognition systems. In contrast, an appearance-based approach does not require any prior knowledge of an object. The latter approach is suitable for the algorithms such as simultaneous localization and mapping which deals with unknown environments [59].

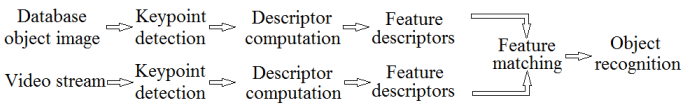


Figure 4.3: Object recognition

Several approaches are proposed for appearance based object recognition.

Santos et al. present the support vector machine (SVM) learning technique as an option to perform appearance-based object recognition [60]. Itti et al. propose the saliency based region selection strategy that extracts multi-scale image features to find salient objects in a cluttered natural scene [61]. Lowes Scale Invariant Feature Transform (SIFT) features [35] provide invariance to change in rotation, scale and viewpoint and are successfully used in object recognition.

4.2.1 Speeded Up Robust Feature (SURF) Extraction and Matching

SURF features are proposed in [37] and they are exploited in various object recognition algorithms. SURF descriptor represents a distribution of Haar wavelet responses within interest point neighbourhood. It is based on the Hessian matrix and relies on integral images to reduce the computation time. In [37] three different versions of the descriptors have been examined and compared with the SIFT descriptor: the standard SURF descriptor, which has a dimension of 64, the extended SURF which has a dimension of 128 and U-SURF version that is not invariant to rotation and has a length of 64 elements. According to the results of the performances for 3 different versions it is indicated that while SURF, extended SURF and upright SURF (U-SURF) extraction processes take 354ms, 391ms, 255ms computational time respectively, SIFT feature extraction method takes 1036ms. In a comparison between the performance of SURF and SIFT feature extraction methods, it is shown that for a scene requiring about 1000ms with SIFT, the extraction of the SURF features takes about 250ms, meaning that the time is reduced by a factor of 4 [62]. Because of the fact that SURF features are not only scale

and rotation invariant but also offer the advantage of being computed very efficiently compared other feature extraction methods, in this work we utilize SURF features in our object recognition algorithm. Extracted SURF features in a catadioptric image are shown in Figure 4.4 and 252 SURF keypoints are extracted.

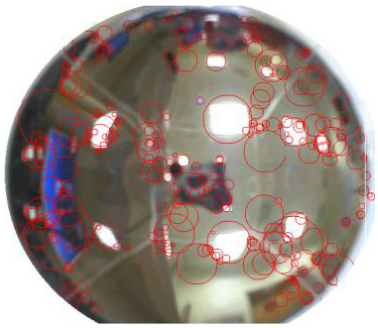


Figure 4.4: Extracted SURF features

Once the SURF keypoints are detected in both database object image and video frames the nearest neighbour matching algorithm is implemented between SURF keypoints. A keypoint in the test image is compared to a keypoint in the database object image by calculating the Euclidean distance between their descriptor vectors. In our work we use SURF descriptor vectors that have lengths of 64 elements. After detection of a matching pair it is examined that the distance is closer than 0.7 times the distance of second nearest neighbour.

4.3 Vehicle Classification via Place Recognition

In this section, we describe the place recognition algorithm to classify the under frames of the vehicles and utilize appearance based mapping algorithm in [14]. This important work proposes an appearance based probabilistic

solution for many problems such as loop closure and perceptual aliasing in SLAM that cannot be solved easily using standard EKF.

To recognize places, the world is modelled as a set of discrete locations and each location is described by a probability distribution over appearance words. Extracted features from images are converted into a bag-of-words representation and a vocabulary is generated. Also, for each location, observation probability of coming from a place in the map or not is examined.

4.3.1 Bag of Words Model

In bag-of-words model, an image is represented as a sort of document, and it contains a set of local descriptors. In order to obtain visual words from images the feature space of the descriptors must be quantized. Thus, a new descriptor vector can be held in terms of the discretized region of feature space to which it belongs. Then, the vocabulary that includes collection of words is generated collecting a large sample of features from a representative corpus of images and quantizing the feature space according to their statistics. In [63], Sivic and Zisserman propose quantizing local image descriptors for the sake of rapidly indexing video frames with an inverted file. They show that local descriptors extracted from features can be mapped to visual words by computing prototypical descriptors with k-means clustering, and that having these tokens enabled faster retrieval of frames containing the same words. Once the descriptor vectors are quantized into visual words, weighting and indexing processes are applied to the vector model as follows: In a vocabulary which includes k words, each document is represented by a k-

vector $V_d = (t_1, ..., t_i, ...t_k)^T$ of weighted frequencies with components

$$t_i = \frac{n_{id}}{n_d} log \frac{N}{n_i} \tag{4.5}$$

where n_{id} is the number of occurences of word i in the document d, n_d is the total number of words in the document d, n_i is the number of occurences of term i in the database and N is the number of documents in the whole database. The weighting is the multiplication of the word frequency and the inverse document frequency. The word frequency weighs words occuring often in a particular document, while the inverse document frequency downweighs words that appear often in database [63]. All of these steps are applied before actual retrieval, and the set of vectors representing all the documents in a corpus are organized as an inverted file. An inverted file index is almost the same as an index in a book, where the keywords are mapped to the page numbers where those words are used. In the visual word case we have, we have a table that points from the word number to the indices of the database images in which that word occurs. Retrieval via the inverted file is faster than searching every image, assuming that not all images include every word. In this work, we utilize SURF features and descriptors to have a bag-of-words representation.

4.3.2 Loop Closure Detection

Detection of loop closure requires the capability of recognizing a previously visited place from current visual sensor measurements. To make it clear one can consider the following illustrative example: Suppose that you are making camp on an island you have not visited before. You would like to discover the

camping environment. At the beginning you mentally keep track of the path you travelled but after some time it would be a challenging work for you to remember in what point you are with respect to the camping area. Instead, if you follow a circular path you will pass the places that you have visited before. Thus, recognizing previously visited places will allow you to estimate your trajectory and where you are with respect to the camping area.

In this work, we address the problem of loop closure detection as an image retrieval task. To classify the vehicles, a newly visited place is examined if it is a new under frame or an old one that is seen before. To achieve this aim $P(Z_k \mid Z^{k-1})$ is calculated which stands for the probability of an observation at a particular sample time k, given the observations till sample time $k-1$. For the theory of loop closure detection interested readers may refer to [24].

4.3.3 Loop Closure Probability Calculation

To calculate $p(Z_k \mid Z^{k-1})$ explicitly, the world is divided into the set of mapped places M and the unmapped places $\bar{M}$ [24]:

$$p(Z_k \mid Z^{k-1}) = \sum_{m \in M} p(Z_k \mid L_m) p(L_m \mid Z^{k-1}) + \sum_{u \in \bar{M}} p(Z_k \mid L_u) p(L_u \mid Z^{k-1}) \tag{4.6}$$

where $p(Z_k \mid L_m)$ and $p(Z_k \mid L_u)$ are observation likelihoods, and $p(L_m \mid Z^{k-1})$ and $p(L_u \mid Z^{k-1})$ are prior beliefs. The second summation cannot be evaluated directly because it involves all possible unknown places. Hence, it is approximated via sampling. The procedure is to sample location models L_u according to the distribution by which they are generated by the environment, and to evaluate $\sum_{u \in \bar{M}} p(Z_k \mid L_u) p(L_u \mid Z^{k-1})$ for the sampled location

models. In order to do this, some method of sampling location models L_u is required. An observation Z is sampled and used to create a place model. In general, this sampling procedure will not create location models according to their true distribution because models may have been created from multiple observations of the location [24]. However, it will be a good approximation when the robot is exploring a new environment, where most location models will have only a single observation associated with them. Having sampled a location model, one must evaluate $p(Z_k \mid L_u)p(L_u \mid Z^{k-1})$ for the sample. The prior probability of the sampled space model with respect to history of observations $p(L_u \mid Z^{k-1})$ is assumed to be uniform over samples, and as a result Eq. 4.6 becomes:

$$p(Z_k \mid Z^{k-1}) \approx \sum_{m \in M} p(Z_k \mid L_m)p(L_m \mid Z^{k-1}) + p(L_{new} \mid Z^{k-1}) \sum_{u=1}^{n_s} \frac{p(Z_k \mid L_u)}{n_s} \tag{4.7}$$

where n_s is the number of samples used, and $p(L_{new} \mid Z^{k-1})$ is prior probability of being at a new place [24].

Chapter V

5 Simulation and Experimental Results

In this chapter simulation results that are performed for Chapter 3 and the experimental results for Chapter 4 are provided, respectively.

5.1 Simulation Results

The performance of the technique which is proposed to improve the accuracy of vSLAM algorithm is verified with simulation results. Ramp and circular inputs are used to generate the odometry data. Odometry and camera outputs are fused in EKF to estimate states of the mobile robot. Extended Kalman filter both estimates the mobile robot states and generates the map of the unknown environment. Inputs for the system are summarized in Table 5.1.

In this table x_r, y_r, θ_r indicate the reference pose of the mobile robot and v_r, w_r denote reference linear and angular velocities of the mobile robot, respectively. Simulation results for the ramp trajectory is depicted for 120 seconds, and 1/50 is chosen for sampling time both for EKF and the camera. In Figure 5.1 (a), (b) and (c) robot pose estimation is shown for ramp input. According to the Figure 5.1 (a) and (b), x and y positions of the mobile robot increase as time increases. Given the control input that is shown in Table 1 for ramp input, x position coordinate of the mobile robot increases

Table 5.1: System Inputs

Type of Input	Input
Ramp Trajectory	$v_r = 0.3[m/s]$
	$w_r = 0[rad/s]$
	$\theta_r = w_r t[rad]$
	$x_r = v_r t[m]$
	$y_r = 0.09t + 0.7[m]$
Circular Trajectory	$v_r = 0.3[m/s]$
	$w_r = 0.6[rad/s]$
	$\theta_r = w_r t[rad]$
	$x_r = x_0 + 5sin(\theta_r)[m]$
	$y_r = y_0 - 5cos(\theta_r)[m]$
	$x_0 = 2[m]$
	$y_0 = 2[m]$

more rapidly than y coordinate position. Initial robot pose as well as the initial camera frame are used as the reference coordinate system and all estimates are represented with respect to this frame. In Figure 5.1 (c), θ, heading angle estimation is shown. When mobile robot starts to navigate in the environment, it has a rotation at the beginning of the movement for trajectory tracking that is related to the ramp control input. The errors between reference and estimated pose states are less than 1%.

In Figure 5.2 (a), (b) and (c) pose estimation of the NWMR is shown for the circular trajectory. The simulation for circular trajectory is performed for 30 seconds and 1/50 sampling time is chosen again for both EKF and the camera as in the ramp input. In Figure 5.2 (a) and (b), x and y position estimates are depicted. Given constant linear and angular velocity inputs, 0.3 [m/s] and 0.6 [rad/s] respectively, mobile robot navigates in circular trajectory in the environment.

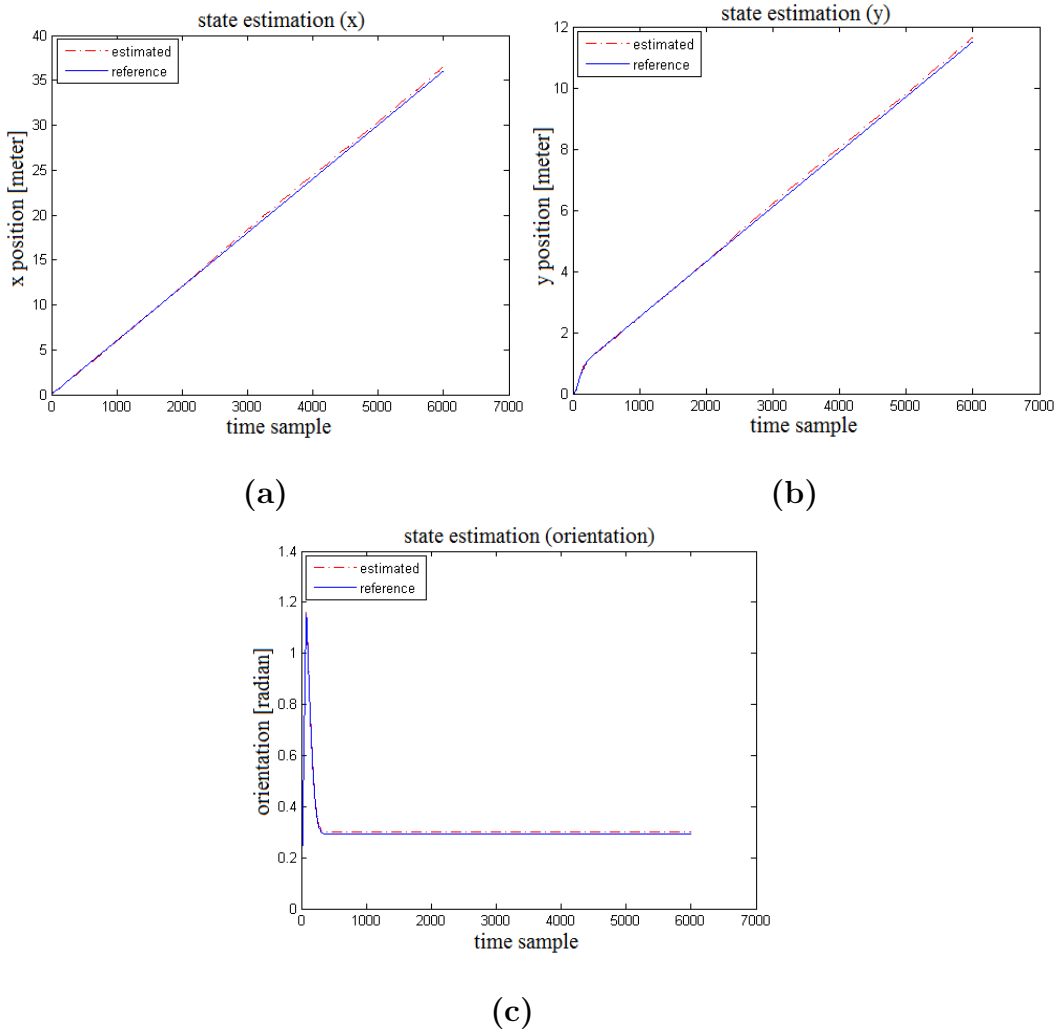


Figure 5.1: x, y and θ state (pose) estimations by EKF for ramp input

In the graph which is shown in Figure 5.2 (a), at 800 and 1300 time samples, there occurs some differences between reference and estimated states. The reason why these differences occur is the rapid increase in heading angle and hence decrease in the overlap area in consecutive image frames. Reduction in the overlapped area between consecutive frames gives rise to decrease

in stable feature point extraction and consequently higher noise in map building.

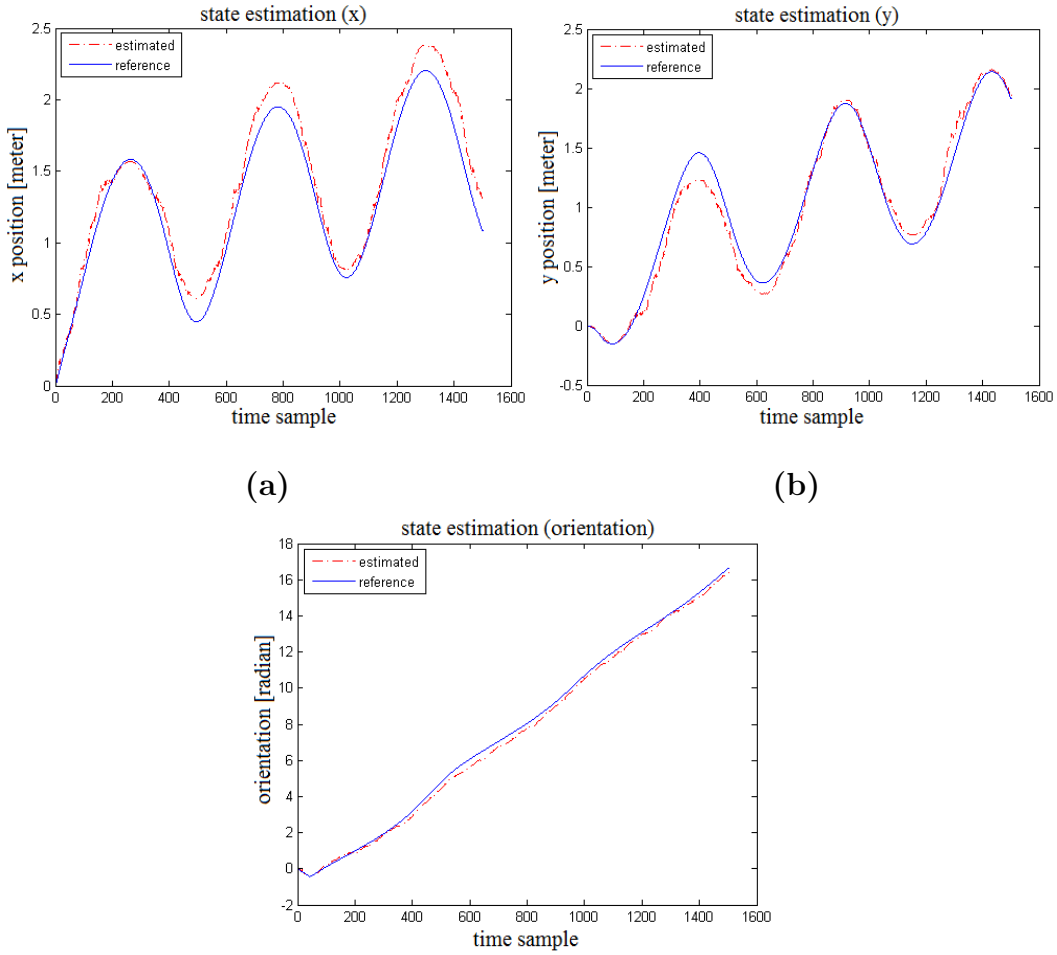
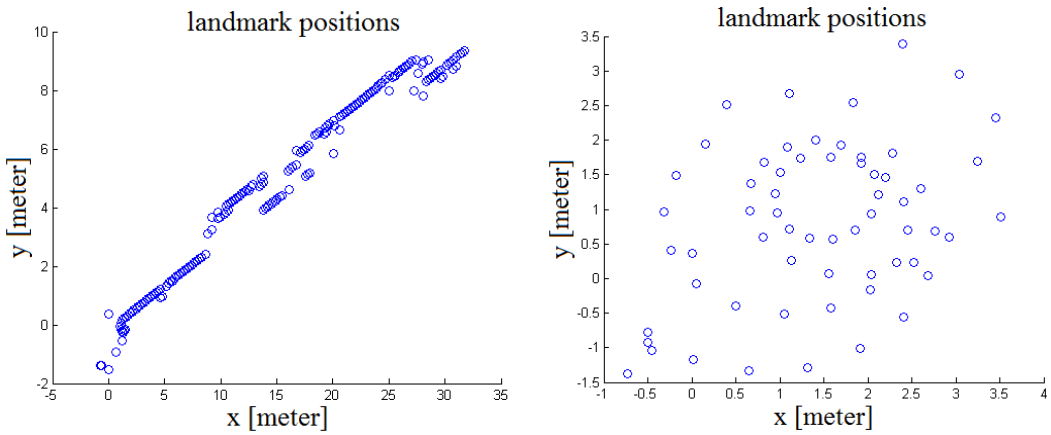


Figure 5.2: x, y and θ state (pose) estimations by EKF for circular input

In our vSLAM algorithm the accuracy of the mobile robot localization is highly dependent on the map building. Errors in these regions are approximately 8%. However, between 800 and 1300 time samples, the reference and

the estimated states are very close to each other, i.e. the error rate is below 1%. This promising result is obtained thanks to the stabilized extracted feature points and validates the performance of our proposed algorithm. In Figure 5.2 (c), it is seen that heading angle increases continuously with time.

The most prominent result of the proposed technique is the accuracy improvement of visual simultaneous localization and map building algorithm using stabilized feature point extraction. Subsequent video frames are stabilized and Harris corner features are extracted from stabilized video sequences. In Figure 5.3 (a) and (b), landmark positions for ramp and circular inputs are shown.



(a) **(b)**

Figure 5.3: Landmark positions: (a) ramp trajectory, (b) circular trajectory

While mobile robot is travelling in the unknown environment with given control inputs, naturally located planar landmarks are extracted and used for measurement update in EKF. In vSLAM algorithms, generating consistent map is one of the most crucial processes to obtain accurate navigation results. Acquiring these naturally located features in a consistent way by neglecting

unwanted camera motion and jitter, our technique builds a consistent map and improves the localization correctness as shown in Figures 5.1 and 5.2.

5.2 Experimental Results

We verified our under vehicle perception algorithm with experimental work. Our proposed solution is implemented using a non-holonomic mobile robot in a laboratory environment. In our implementations the bottom of the tables in the laboratory are considered as the under vehicles. A database that includes eight different under vehicle images is used in this experimental work. They are attached to the bottom of the tables and the mobile robot navigates under the tables. All the algorithms are implemented in Microsoft Visual C++ and OpenCV 2.4.4 that are installed to the on-board computer mounted to the body of the mobile robot.

In general, the mapping from world points to image pixels is non-linear. Normally a correction would be needed for the imaging model. Matching of features in viewpoint change would be quite challenging. The planar motion assumption eliminates the requirement of correction for the projection. For example, a lens distortion correction is not used to project the straight lines.

In object recognition algorithm an appearance based method is used. The features are matched between the catadioptric video sequences and database perspective images. There is no correction for the catadioptic image mapping from world points to image pixels since the appearance based approach is employed. Moreover, an appearance based mapping approach is exploited to classify the vehicles. Extracted features from catadioptric images are converted into the bag of words representation. These images are then utilized in the construction of the topologic map of the environment.

5.2.1 Experimental Setup

The mobile robot that is used in our experimental work includes a processor, an on-board computer, a catadioptric camera system and a rechargeable lithium polymer battery (Figure 5.4). Working principle of the experimental setup is depicted in Figure 5.5. The power of both of the processor and the computer are supplied via 14.8 V lithium polymer batteries. The processor is inserted to the mobile robot body for sending the control commands to the wheels of the mobile robot. Philips USB camera with a catadioptric mirror is connected to the on-board computer. The communication between the processor of the mobile robot and computer is provided using RS-232 communication protocol. A network is established between the on-board computer and a laptop and shown with a dashed line in Figure 5.5. The laptop is used as an external device to display the camera results on the screen.

5.2.2 Results for Object Recognition

While the mobile robot navigates under the tables it starts to monitor under vehicle images that are attached to the bottom of the tables. In a certain image, we attach a test object which is one of the database images in the mobile robot and make the mobile robot detect the object. In this implementation we use the SURF features for extraction and matching between the object and the video frames, and Random Sample Consensus (RANSAC) algorithm to neglect the matches that are found as outliers. If the mobile robot detects the object in a catadioptric image it is shown using a line as in Figure 5.6.

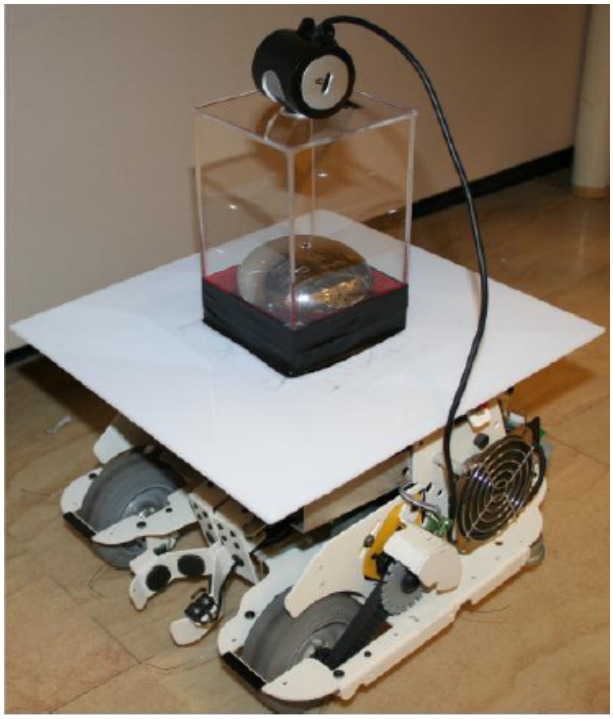


Figure 5.4: Experimental setup

5.2.3 Results for Vehicle Classification

In the first experiment, to show the accuracy of the FAB-MAP algorithm we use a hand-held perspective camera with taking under frame images of the vehicles. Seven different under frame images of the vehicles are used to calculate the resultant confusion matrix shown in Figure 5.7. When a new place is seen, the relevant diagonal element of the matrix is assigned with a high probability value and this element is depicted bright in the matrix. Regarding the loop closure detection, off-diagonal elements of the matrix are used and indicated bright on the off-diagonal region.

In Figure 5.7, it is seen that all of the diagonal elements of the matrix are bright and it is understood that all of the visited places are new and there is

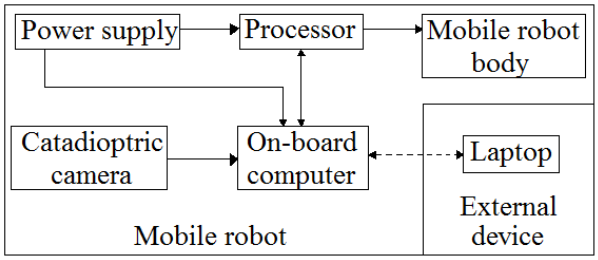


Figure 5.5: Working principle of the experimental setup

Figure 5.6: Detected objects

no loop closure detection. Namely each of these images belongs to different under vehicles and they can be classified in seven groups. The probability of being a new vehicle under frame for the third one is 0.995 whilst the fifth one is 0.996968.

Once we obtain this resultant confusion matrix we try a different set of images to show the loop closure detection. The relevant loop closure detections are shown in Figure 5.8. In Figure 5.8 (a) and (b), two different images of the same vehicle under frame for the first and ninth places are shown whilst in Figure 5.8 (c) and (d) the same under vehicle images are depicted for the third and tenth places (see Figure 5.9). While the loop closure probability for the ninth and first images is 0.961524, the probability of being a new place for the ninth image is 0.0150896. Similarly, loop closure probability for the third and tenth images is 0.954927 and assigned probability for being a

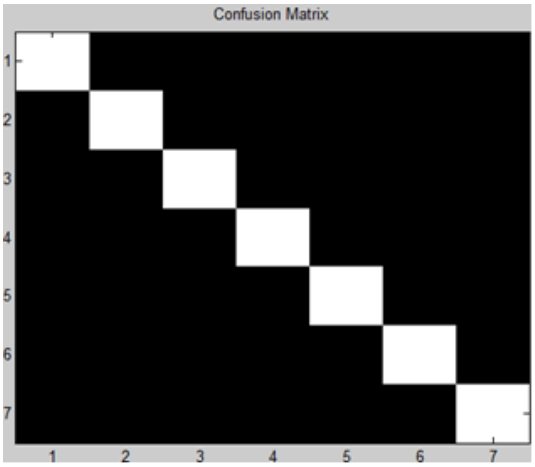


Figure 5.7: Confusion matrix: all visited places are seen first

new place for the tenth image is 0.00117. These ten vehicles can be classified under eight groups because of detecting two loop closures in the ninth and tenth steps. These loop closure results allow us to classify the vehicles merely using their under frames. In the experiment we use Open FAB-MAP software released by Glover et al. [64].

In the second experiment, we take six different under frame images of the vehicles using our proposed catadioptric camera system mounted to the body of the mobile robot. Some example images are shown in Figure 5.10. As it is seen from Figure 5.10, a different place is assigned for each different vehicle under frame. Firstly, we capture the omnidirectional images of the six consecutive different under vehicles and related confusion matrix is depicted in Figure 5.11. Because of the fact that all images are different, the diagonal elements of the matrix are indicated bright with high probability that explains the related visited place is newly seen. For example, the probability of being a new under frame for third place is 0.996 and for the sixth place is 0.997.

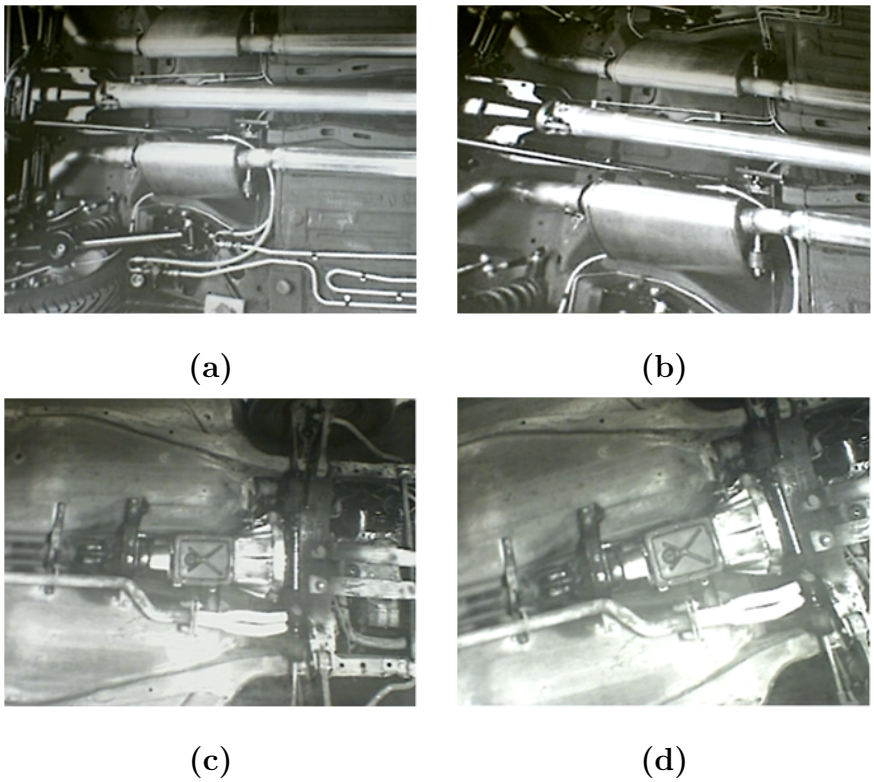


Figure 5.8: Loop closure detections: between (a) and (b) for the ninth and first places and between (c) and (d) for the tenth and third places

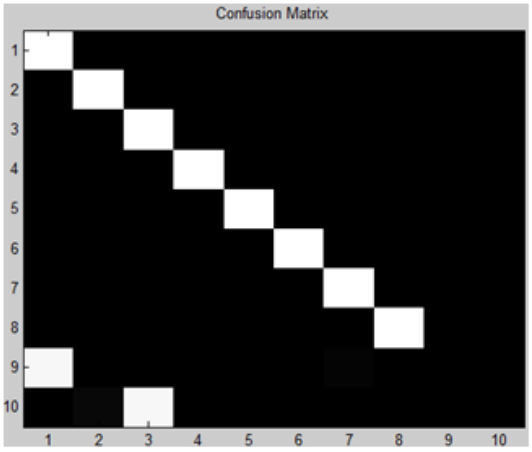


Figure 5.9: Confusion matrix: loop closures between the ninth and first places, and between the tenth and third places

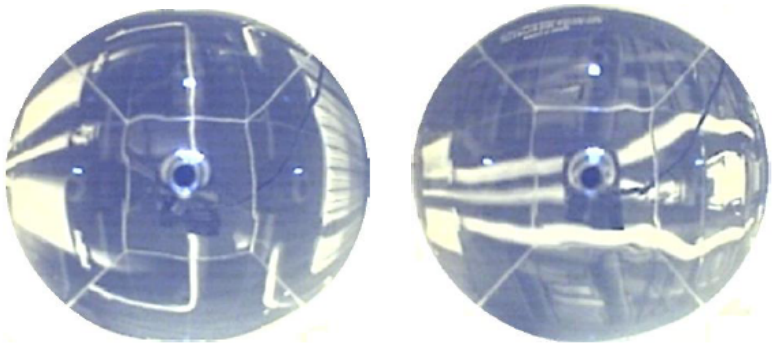


Figure 5.10: Omnidirectional images of under vehicles

Then, we deliberately enlarge the database by two additional images that belong to the same under vehicles in the database. This time, the resultant confusion matrix is shown in Figure 5.12 that explains the loop closures between the fourth and seventh places and first and eighth places.

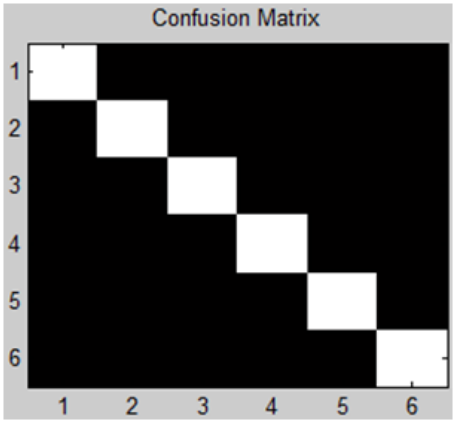


Figure 5.11: Confusion matrix for omnidirectional images: all visited places are seen first

While the loop closure probability for the forth and seventh images is 0.9742, the probability of being a new place for the seventh image is 0.01296.

Similarly, loop closure probability for the first and eighth images is 0.96289 and assigned probability for being a new place for the eighth image is 0.0023. These ten vehicles can be classified under six groups because of detecting two loop closures in the seventh and eighth steps.

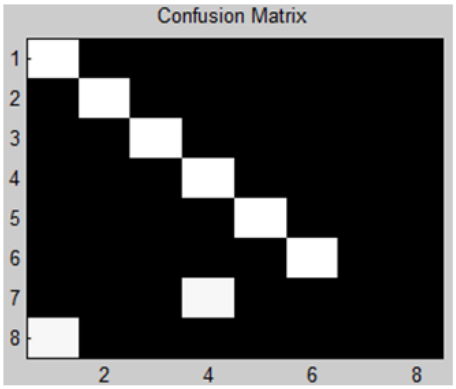


Figure 5.12: Confusion matrix for omnidirectional images: loop closures in the seventh and eighth places

Chapter VI

6 Conclusion and Future Work

We have provided a detailed analysis of SLAM for the navigation of an autonomous mobile robot. Firstly, we described the probabilistic formulation of the SLAM problem. We then reviewed different kinds of SLAM algorithms. In particular, fusion of different sensors in EKF for the SLAM problem usually provides consistent map building and localization. Appearance based mapping algorithm was then elaborated for large scale SLAM problems. With feature extraction, matching and tracking, we developed a good insight into the vSLAM problem.

We proposed an improvement technique for the accuracy of the metric based vSLAM algorithm. We incorporated video stabilization into vSLAM for feature extraction, and consequently for map building and localization. In vSLAM, the performance of the algorithm depends on both the accuracy of the map and localization of the robot. It has been shown that consistent feature extraction technique both improves the accuracy of map building and localization of the mobile robot by neglecting unwanted sensor motion and the noises that are caused by the external factors.

We then described under vehicle surveillance for high level safety measures using a catadioptric camera system. We used object detection algorithm to recognize hidden objects mounted to the under frames of the vehicles and

exploited FAB-MAP algorithm to classify the vehicles. The imaging model of the catadioptric system is developed to indicate its advantages over standard perspective cameras. A vehicle equipped with a catadioptric system can see not only the frontal direction but also the right, left and back sides thanks to the large field of view. The number of extracted features from single catadioptric image is higher than a perspective image and so matching between two consecutive images taken from catadioptric cameras gives rise to much more consistent results in terms of object recognition and classification, localization and mapping.

In MATLAB/Simulink simulations, we verified the accuracy of the proposed solution for the improvement of the vSLAM problem. Ramp and circular control inputs were used to generate the odometry data. When a mobile robot navigates through an unknown environment with given control inputs, naturally located planar landmarks are extracted and used for measurement update in EKF. Landmark positions for ramp and circular inputs are shown in the simulation results. Odometry and camera outputs are fused in EKF to estimate states of the mobile robot. Also, the estimation results of the robot pose are shown separately for the ramp and circular inputs. In both cases, it is shown that errors between reference and estimated pose states are less than 1%.

In experimental work, we presented the feasibility of the proposed method in a laboratory environment using a non-holonomic wheeled mobile robot equipped with a catadioptric camera. In our implementations the bottom of the tables in the laboratory are considered as under vehicles. Algorithms are implemented in Microsoft Visual C++ and OpenCV 2.4.4 that are installed to the on-board computer mounted to the body of the mobile robot. To

demonstrate the object recognition, we attached a test object which was one of the database images in the mobile robot and made the mobile robot detect the object. In this implementation we used the SURF features for extraction and matching between the object and the video frames, and Random Sample Consensus (RANSAC) algorithm to neglect the matches that are found as outliers. To verify the classification of the vehicles, a hand-held perspective camera is utilized in the first experiment. Seven different under frame images of the vehicles were used to calculate the confusion matrices which show the loop closures are detected or not. Once the classification results were verified by a perspective camera, in a separate experiment, we took six different under frame images of the vehicles using our proposed catadioptric camera system mounted to the body of the mobile robot and the classification results were reported.

Because of having a small under vehicle image database, in this thesis we showed the feasibility of the proposed under vehicle perception method for relatively small scale perception tasks. As a future work, the proposed method can be extended to large scale under vehicle perception missions. A more extended under vehicle image database should be used for the classification of the vehicles and detection of the hidden objects. Also, a vision-based control can be applied to the wheeled mobile robot utilizing extracted features from the environment.

References

[1] H.D. Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *Robotics and Automation Magazine*, 13:99–108, 2006.

[2] Y.J. Lee and S. Sung. *Vision based SLAM for mobile robot navigation using distributed filters.* Book Chapter, 2010.

[3] N. Ayache and O. Faugeras. Building, registrating, and fusing noisy visual maps. *Int. Jornal of Robotics Research*, 7:45–65, 1988.

[4] J.J. Leonard and H.D. Whyte. Simultaneous map building and localization for an autonomous mobile robot. *International Workshop on Intelligent Robots and Systems*, 3:1442–1447, 1991.

[5] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *Transactions on Robotics and Automation*, 17:242–257, 2001.

[6] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainity using scale-invariant visual landmarks. *The International Journal of Robotic Research*, 21:735–758, 2002.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, 2002.

[8] F. Fang, X. Ma, and X. Dai. A multisensor fusion SLAM approach for mobile robots. *International Conference on Mechatronics and Automation*, 4:1837–1841, 2005.

[9] W.L.D. Lui and R. Jarvis. A pure vision based approach to topological SLAM. *International Conference on Intelligent Robots and Systems*, pages 3784–3791, 2010.

[10] Z. Chen, J. Samarabandu, and Rodrigo. R. Recent advances in simultaneous localization and map-building using computer vision. *Advanced Robotics*, 21:233–265, 2007.

[11] S. Lacroix, A. Mallet, I.K. Jung, T. Lemaire, and J. Sola. *Vision based SLAM.* Oxford, 2006.

[12] C. Tomasi and T. Kanade. *Detection and tracking of point features.* Technical Report CMU, 1991.

[13] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainity. *The International Journal of Robotic Reseach*, 5:56–68, 1986.

[14] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. *Proceedings of the 4th International Symposium on Robotics Research*, pages 467–474, 1988.

[15] S. Jia and A. Yasuda. *Mobile robot localization and map building for a nonholonomic mobile robot.* Book Chapter, 2009.

[16] N. Karlsson and et al. The vSLAM algorithm for robust localization and mapping. *International Conference on Robotics and Automation*, pages 24–29, 2005.

[17] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM:

Real-time single camera SLAM. *Transaction on Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007.

[18] J. Aulinas, Y. Petillot, J. Salvi, and X. Llado. The SLAM problem: A survey. *Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pages 363–371, 2008.

[19] S.C. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *International Symposium Aerospace/Defense Sensing, Simulation and Controls*, 3068:182–193, 1997.

[20] S.J. Julier. The spherical simplex unscented transformation. *Proceedings of the American Control Conference*, 3:2430–2434, 2003.

[21] N. Sunderhauf, S. Lange, and P. Protzel. Using the unscented Kalman filter in mono SLAM with inverse depth parametrization for autonomous airship control. *International Workshop on Safety, Security and Rescue Robotics*, pages 1–6, 2007.

[22] M. Cummins. *Probabilistic localization and mapping in appearance space*. PhD Thesis, 2009.

[23] E.B.B. Cortes. *Appearance based mapping and localization using feature stability histograms for mobile robot navigation*. PhD Thesis, 2012.

[24] M. Cummins and P. Newman. FABMAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27:647–665, 2008.

[25] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Appearance-

only SLAM at large scale with FAB-MAP 2.0. *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, 2002.

[26] R. Szeliski. *Computer vision: Algorithms and applications*. Springer, 2010.

[27] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[28] B.D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. *Seventh International Joint Conference on Artificial Intelligence*, 17:674–679, 1981.

[29] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[30] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37:151–172, 2000.

[31] C. Harris and M.J. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–152, 1988.

[32] B. Triggs. Detecting keypoints with stable position, orientation, and scale under illumination changes. *Eighth European Conference on Computer Vision*, pages 100–113, 2004.

[33] P. Anandan. Computing dense displacement fields with confidence measures in scenes containing occlusion. *Image Understanding Workshop*, pages 236–246, 1984.

[34] J. Shi and C. Tomasi. Good features to track. *Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[35] D.G. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[36] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[37] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110:346–359, 2008.

[38] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *Transaction on Robotics and Automation*, 17:242–257, 2001.

[39] F. Kühne, J.M.G. Jr da Silva, and W.F. Lages. Model predictive control of a mobile robot using input-output linearization. *Proceedings of Mechatronics and Robotics*, 2005.

[40] W.F. Lages and J.A.V. Alves. Real-time control of a mobile robot using linearized model predictive control. *Proc. of 4th IFAC Symposium on Mechatronics Systems*, 4:968–973, 2006.

[41] Y.T. Wang, Y.C. Feng, and D.Y. Hung. Detection and tracking of moving objects in SLAM using vision sensors. *Trans. Instrumentation and Measurement Technology Conference*, 4:1–5, 2011.

[42] J.J. Craig. *Introduction to Robotics : Mechanics and Control*. Addison-Wesley, 1989.

[43] S. Haykin. *Kalman Filtering and Neural Networks*. John Wiley and Sons Inc, 2001.

[44] R.A. Hamzah, R.A. Rahim, and Z.M. Noh. Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application. *Comp. Science and Information Technology*, 1:652–657, 2010.

[45] S. Baker and S.K. Nayar. A theory of single-viewpoint catadioptric image formation. *Int. Journal of Computer Vision*, 35:1–22, 1999.

[46] R. Benosman and S.E. Kang. *Panoramic Vision: Sensors, Theory, and Applications*. Springer-Verlag New York, 2001.

[47] C. Geyer and K. Daniilidis. Structure and motion from uncalibrated catadioptric views. *Int. Conf. on Computer Vision and Pattern Recognition*, pages 279–286, 2001.

[48] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *Int. Journal of Computer Vision*, 49:23–37, 2002.

[49] B. Micusik and T. Pajdla. Autocalibration and 3D reconstruction with noncentral catadioptric cameras. *Int. Conf. on Computer Vision and Pattern Recognition*, 1:58–65, 2004.

[50] M. Schönbein, B. Kitt, and M. Lauer. Environmental perception for intelligent vehicles using catadioptric stereo vision systems. *Proc. of the European Conference on Mobile Robots (ECMR)*, pages 1–6, 2011.

[51] W.L.D. Lui and R. Jarvis. Eye-full tower: A GPU based variable multi-baseline omnidirectional stereovision with automatic baseline selection for outdoor mobile robot navigation. *Robotics and Autonomous Systems*, 58:747–761, 2010.

[52] T. Gandhi and M. Trivedi. Vehicle surround capture: Survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps. *Trans. on Intelligent Transportation Systems*, 7:293–308, 2006.

[53] M. Schönbein, M. Rapp, and M. Lauer. Panoramic 3D reconstruction with three catadioptric cameras. *Advances in Intelligent Systems and Computing*, 193:345–353, 2013.

[54] P. Dickson and et al. Mosaic generation for under vehicle inspection. *Proc. of Sixth IEEE Workshop on Applications of Computer Vision*, pages 251–256, 2002.

[55] S.R. Sukumar, D.L. Page, A.V. Gribok, A.F. Koschan, and M.A. Abidi. Robotic three dimensional imaging system for under vehicle inspection. *Journal of Electronic Imaging*, 15, 2006.

[56] C.N. Anagnostopoulos, I. Giannoukos, T. Alexandropoulos, A. Psyllos, V. Loumos, and E. Kayafas. Integrated vehicle recognition and inspection system to improve security in restricted access areas. *Annual Conference on Intelligent Transportation Systems*, pages 1893–1898, 2010.

[57] E.E. Ruiz and K.L. Head. Use of an automatic under vehicle inspection system as a tool to streamline vehicle screening at ports of entryand

security checkpoints. *European Intelligence and Security Informatics Conference*, pages 329–333, 2012.

[58] C. Mei, S. Benhimane, E. Malis, and P. Rives. Homography-based tracking for central catadioptric cameras. *Int. Conf. on Intelligent Robots and Systems*, pages 669–674, 2006.

[59] Y.J. Lee and J.B. Song. Visual SLAM in indoor environments using autonomous detection and registration of objects. *Int. Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 671–676, 2008.

[60] E.M.D. Santos and H.M. Gomes. Appearance-based object recognition using support vector machines. *Computer Graphics and Image Processing*, page 399, 2001.

[61] L. Itti, C. Koch, and E. Neibur. A model of saliency-based visual attention for rapid scene analysis. *Trans. on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

[62] V. Högman. *Building a 3D map from RGB-D sensors.* Master Thesis, 2012.

[63] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *Int. Conf. on Computer Vision*, 2:1470–1477, 2003.

[64] A. Glover, W. Maddern, S. Reid, M. Milford, and G. Wyeth. OpenFABMAP: An open source toolbox for appearance based loop closure detection. *International Conference on Robotics and Automation*, pages 4730–4735, 2012.